

# Game-theoretic Model of Computation

Norihiro Yamada

norihiro1988@gmail.com

University of Oxford

February 23, 2017

## Abstract

We introduce in the present paper an *intrinsic* notion of “(effective) computability” in game semantics motivated by the fact that strategies in game semantics have been defined *recursive* if they are “computable in an *extrinsic* sense”, i.e., they are representable by partial recursive functions, and so it has been difficult to regard game semantics as an *autonomous* foundation of computation. As a consequence, we have formulated a general notion of “algorithms” under the name of *effective strategies*, giving rise to a mathematical model of computation in the same sense as Turing machines but beyond computation on natural numbers, e.g., *higher-order* one, *solely in terms of games and strategies*. It subsumes computation of the programming language PCF, and so it is in particular *Turing complete*. Notably, effective strategies have a natural notion of *types* (i.e., games) unlike Turing machines, while they are *non-inductively* defined as opposed to partial recursive functions as well as *semantic* in contrast with  $\lambda$ -calculi and combinatory logic. Thus, in a sense, we have captured a mathematical (or semantic) notion of computation (and computability) that is more general than the “classical ones” in a fundamental level. Exploiting the flexibility of game semantics, our game-theoretic model of computation is intended to give a mathematical foundation of various (constructive) logics and programming languages.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Preliminaries: dynamic games and strategies</b>	<b>3</b>
2.1	On the tags for disjoint union of sets . . . . .	4
2.2	Dynamic games . . . . .	4
2.3	Dynamic strategies . . . . .	12
<b>3</b>	<b>Effective strategies</b>	<b>13</b>
3.1	Effective strategies . . . . .	13
3.2	Examples of atomic strategies . . . . .	23
3.3	Turing completeness . . . . .	34
<b>4</b>	<b>Conclusion and future work</b>	<b>35</b>

# 1 Introduction

*Game semantics* [A<sup>+</sup>97, AM99, Hyl97] refers to a particular kind of *semantics of logics and programming languages* in which types and terms are interpreted as *games* and *strategies*, respectively. Historically, game semantics gave the first syntax-independent characterization of the programming language PCF [AJM00, HO00, Nic94]; since then a variety of games and strategies have been proposed to model various programming features [Abr14, AM99]. An advantage of game semantics is this flexibility: It models a wide range of languages by simply varying constraints on strategies, which enables one to compare and relate different languages ignoring superfluous syntactic details. Its another characteristic is its conceptual naturality: It interprets syntax as dynamic interactions between *Player* and *Opponent* of a game, providing an intensional explanation of syntax in a natural and intuitive (yet mathematically precise) manner.

However, although game semantics has provided a unified framework to model various logics and programming languages, it has never been formulated as a *mathematical model of computation* in its own right in the same sense as *Turing machines* [Tur36, Koz12], the  $\lambda$ -calculus [Chu36, Chu40, B<sup>+</sup>84], *combinatory logic* [Sch24, Cur30], etc. More specifically, “(effective) computability” in game semantics has been always *extrinsic* [Abr14]: A strategy has been defined to be *recursive* if it is representable by a partial recursive function [AJM00, HO00]. This is mainly because a primal focus of the field has been *full abstraction* [Win93, Gun92], i.e., to characterize an *observational equivalence* in syntax in a syntax-independent manner; thus, it has not been concerned that much with (*step-by-step*) *processes* of computation. Nevertheless, it is unsatisfactory from a foundational point of view as it does not give much new insight on the notion of “effective computation”. Also, it raises an intriguing mathematical question in its own right: Is there any *intrinsic* (in the sense that it does not have recourse to the standard definition of computability) notion of “effective computability” in game semantics that is *Turing complete* (i.e., it contains every *Turing-computable* or *partial recursive* functions [Cut80, RR67])?

Motivated by the above consideration, in this paper we present the notion of *effective strategies* in game semantics defined solely in terms of games and strategies. Roughly, a strategy is *finitary* if its partial function representation that assigns the next Player’s move to a bounded size of partial history of previous moves, called its *table*, is finite, and *effective* if its table is “describable” by a finitary strategy. They give a reasonable notion of “computability” as finitary strategies are clearly “computable”, and so their “descriptions” can be “effectively read off”. Note that they are defined *intrinsically* in the sense stated above. The main idea is to allow strategies to look at only a *bounded* number of previous moves, and describe them by means that is clearly “effectively executable” but more expressive than finite tables, i.e., finitary strategies. This simple notion subsumes computation of the language PCF, and thus it is *Turing complete*, providing a positive answer to the question posted above. As a result, we have formulated a general notion of “algorithms”, which in turn gives rise to a mathematical model of computation in the same sense as Turing machines but beyond computation on natural numbers which we call the *classical computation*.

In hindsight, our game-theoretic model of computation may be seen as “*interactive Turing machines*” since its computation proceeds as an interaction between Player and Opponent, where Turing machines interact with Opponent only once as they just receive an input and produces an output (if it halts) once and for all, and the current position in a game serves as the current “state of mind” for effective strategies. It is this generalization of Turing machines that gives the game-theoretic model of computation an additional flexibility and computational power, inheriting their semantic and non-inductive nature.

To the best of our knowledge, effective strategies are the first intrinsic characterization of computability in game semantics that is Turing complete. Notably, they are *non-inductively*

defined as opposed to partial recursive functions, *semantic* in contrast with  $\lambda$ -calculi and combinatory logic, and equipped with the notion of *types*, i.e., games, unlike Turing machines. Thus, in a sense, we have captured a mathematical notion of “computation” that is more general than the classical one, e.g., *higher-order* one [LN15], in a fundamental level. Therefore exploiting the flexibility of game semantics, our model of computation has a potential to give a mathematical foundation of a wide range of logics and programming languages.

The rest of the paper proceeds as follows. Defining our games and strategies in Section 2, we define effective strategies and show that they may interpret every term in the language PCF in Section 3. Finally, we make a conclusion and propose some future work in Section 4.

► **Notation.** We use the following notations throughout the paper:

- We use bold letters  $\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}$ , etc. for sequences, in particular  $\epsilon$  for the *empty sequence*, and letters  $a, b, c, d, m, n, x, y, z$ , etc. for elements of sequences. We often abbreviate a finite sequence  $\mathbf{s} = (x_1, x_2, \dots, x_n)$  as  $x_1x_2 \dots x_n$  and write  $s_i$  as another notation for  $x_i$ .
- A concatenation of sequences is represented by a juxtaposition of them, but we write  $as$ ,  $tb$ ,  $ucv$  for  $(a)s$ ,  $t(b)$ ,  $u(c)v$ , etc. We sometimes write  $s.t$  for  $st$  for readability.
- We write  $\text{even}(\mathbf{s})$  (resp.  $\text{odd}(\mathbf{s})$ ) if  $\mathbf{s}$  is of even-length (resp. odd-length). For a set  $S$  of sequences, we define  $S^{\text{even}} \stackrel{\text{df.}}{=} \{\mathbf{s} \in S \mid \text{even}(\mathbf{s})\}$  and  $S^{\text{odd}} \stackrel{\text{df.}}{=} \{\mathbf{t} \in S \mid \text{odd}(\mathbf{t})\}$ .
- We write  $\mathbf{s} \preceq \mathbf{t}$  if  $\mathbf{s}$  is a prefix of  $\mathbf{t}$ . For a set  $S$  of sequences,  $\text{pref}(S) \stackrel{\text{df.}}{=} \{\mathbf{s} \mid \exists \mathbf{t} \in S. \mathbf{s} \preceq \mathbf{t}\}$ .
- For a partially ordered set  $P$  and a subset  $S \subseteq P$ ,  $\text{sup}(S)$  denotes the *supremum* of  $S$ .
- $X^* \stackrel{\text{df.}}{=} \{x_1x_2 \dots x_n \mid n \in \mathbb{N}, \forall i \in \{1, 2, \dots, n\}. x_i \in X\}$  for each set  $X$ .
- For a function  $f : A \rightarrow B$  and a subset  $S \subseteq A$ , we define  $f \upharpoonright S : S \rightarrow B$  to be the *restriction* of  $f$  to  $S$ . Also,  $f^* : A^* \rightarrow B^*$  is defined by  $f^*(a_1a_2 \dots a_n) \stackrel{\text{df.}}{=} f(a_1)f(a_2) \dots f(a_n)$ .
- Given sets  $X_1, X_2, \dots, X_n$ , for each  $i \in \{1, 2, \dots, n\}$  we write  $\pi_i : X_1 \times X_2 \times \dots \times X_n \rightarrow X_i$  for the  $i^{\text{th}}$ -*projection function*  $(x_1, x_2, \dots, x_n) \mapsto x_i$ .
- We write  $x \downarrow$  if an element  $x$  is defined and  $x \uparrow$  otherwise.

## 2 Preliminaries: dynamic games and strategies

This section presents our games and strategies. It is essentially the “*dynamic refinement*” of McCusker’s variant [AM99, McC98], which is proposed by the present author and Abramsky in [YA16]. Its main purpose is to refine the composition of strategies as “*non-normalizing composition plus hiding*” in order to capture *dynamics* and *intensionality* in computation. We have chosen this variant since the non-normalizing composition preserves “atomic computational steps” in strategies, and thus effective strategies are closed under it (but not under the usual composition). However, we need a minor modification: A particular implementation of tags for disjoint union of sets of moves (for constructions on games) has to be adopted as manipulations of the tags must be “effectively executable” by strategies, and strategies should behave “consistently” up to permutations of tags in exponential ! as in [AJM00, McC98].

## 2.1 On the tags for disjoint union of sets

In game semantics, we often take disjoint union of sets (of *moves*) when we form compound games such as tensor  $\otimes$ , where we usually treat “tags” for such disjoint union *informally* for brevity [AM99, McC98]. However, since we are concerned with “*effective computability*”, including how to “*effectively*” handle “tags”, we have to formulate them rigorously. For this reason, we introduce:

► **Definition 2.1.1** (Effective tags). An *effective tag* is a finite sequence over the alphabet  $\Sigma = \{\#, |\}$ , where  $\#, |$  are arbitrarily fixed symbols. We write  $\underline{i}$  for  $\underbrace{|| \dots ||}_i$  for each  $i \in \mathbb{N}$ .

► **Definition 2.1.2** (Decoding and encoding). The *decoding function*  $de : \Sigma^* \rightarrow \mathbb{N}^*$  is defined by  $de(\gamma) \stackrel{\text{df.}}{=} (i_1, i_2, \dots, i_k) \in \mathbb{N}^*$  for all  $\gamma \in \Sigma^*$ , where  $\gamma = \underline{i_1} \# \underline{i_2} \# \dots \# \underline{i_{k-1}} \# \underline{i_k}$ , and the *encoding function*  $en : \mathbb{N}^* \rightarrow \Sigma^*$  by  $en(j_1, j_2, \dots, j_l) \stackrel{\text{df.}}{=} \underline{j_1} \# \underline{j_2} \# \dots \# \underline{j_{l-1}} \# \underline{j_l}$  for all  $(j_1, j_2, \dots, j_l) \in \mathbb{N}^*$ .

Clearly, the functions  $de : \Sigma^* \rightleftharpoons \mathbb{N}^* : en$  are mutually inverses (n.b. they both map  $\epsilon$  to itself). In fact, effective tags  $\gamma$  are to represent finite sequences  $de(\gamma)$  of natural numbers.

However, effective tags are not sufficient for our purpose: For “nested exponentials !”, we need to “*effectively*” associate a natural number to each finite sequence of natural numbers in an “*effectively*” invertible way. Of course it is possible as there is a computable bijection  $\langle \_ \rangle : \mathbb{N}^* \rightarrow \mathbb{N}$  whose inverse is also computable by an elementary fact from computability theory [Cut80, RR67], but we cannot rely on it as we are aiming at developing an *autonomous* foundation of “*effectivite computability*”. On the other hand, this bijection is necessary only for manipulating effective tags, and so we would like to avoid an involved mechanism for it.

Our solution for this problem is to simply introduce some symbols to *denote* the bijection:

► **Definition 2.1.3** (Extended effective tags). An *extended effective tag* is an expression  $e \in (\Sigma \cup \{\langle, \rangle\})^*$  generated by the rule  $e \stackrel{\text{df.}}{=} \gamma | e_1 \# e_2 | \langle e \rangle$ , where  $\gamma$  ranges over effective tags.

► **Definition 2.1.4** (Extended decoding). The *extended decoding function*  $ede : \mathcal{T} \rightarrow \mathbb{N}^*$  is defined by  $ede(\gamma) \stackrel{\text{df.}}{=} de(\gamma)$ ,  $ede(e_1 \# e_2) \stackrel{\text{df.}}{=} ede(e_1)ede(e_2)$ ,  $ede(\langle e \rangle) \stackrel{\text{df.}}{=} \langle ede(e) \rangle$ , where  $\mathcal{T}$  is the set of extended effective tags, and  $\langle \_ \rangle : \mathbb{N}^* \rightarrow \mathbb{N}$  is any computable bijection fixed throughout the present paper such that  $\langle i_1, i_2, \dots, i_k \rangle \neq \langle j_1, j_2, \dots, j_l \rangle$  whenever  $k \neq l$  (see, e.g., [Cut80]).

Of course, we lose the bijectivity between  $\Sigma^*$  and  $\mathbb{N}^*$  for extended effective tags, but in return, we may “symbolically execute” the bijection  $\langle \_ \rangle : \mathbb{N}^* \rightarrow \mathbb{N}$  by just inserting  $\langle, \rangle$ . From now on, the word *tags* refers to *extended effective tags*, and we write  $e, f, g, h$ , etc. for tags.

► **Definition 2.1.5** (Tagged elements). A *tagged element* is any pair  $[m]_e \stackrel{\text{df.}}{=} (m, e)$  with  $e \in \mathcal{T}$ .

► **Notation.** We often abbreviate a tagged element  $[m]_e$  as  $m$  if the tag  $e$  is not important.

## 2.2 Dynamic games

Our games are essentially *dynamic games* introduced in [YA16] equipped with an equivalence relation on *positions* that “ignores” permutations of tags in exponential ! as in [AJM00, McC98].

The main idea of dynamic games is to introduce a distinction between *internal* and *external* moves; internal moves constitute “internal communication” between strategies, and they are to be *a posteriori* hidden by the *hiding operation*. Conceptually, internal moves are “invisible” to

Opponent as they represent how Player *internally* calculates the next external move. In this manner, dynamic games provide a “universe of computation” in which *intensionality* and *dynamics* in computation are represented by internal moves and the hiding operation, respectively. We first quickly review their basic definitions; see [YA16] for the details.

As games defined in [AM99, McC98], dynamic games are based on two preliminary concepts: *arenas* and *legal positions*. An arena defines basic components of a game, which in turn induces a set of legal positions that specifies the basic rules of the game.

► **Definition 2.2.1** (Arenas [YA16]). A (*dynamic*) *arena* is a triple  $G = (M_G, \lambda_G, \vdash_G)$ , where:

- $M_G$  is a set of tagged elements called *moves* such that  $\{\pi_1(m) \mid m \in M_G\}$  is finite
- $\lambda_G : M_G \rightarrow \{O, P\} \times \{Q, A\} \times \mathbb{N}$  is a function called the *labeling function*, where O, P, Q, A are arbitrarily fixed symbols, that satisfies  $\sup(\{\lambda_G^{\mathbb{N}}(m) \mid m \in M_G\}) \in \mathbb{N}$
- $\vdash_G \subseteq (\{\star\} \cup M_G) \times M_G$  is a relation, where  $\star$  is an arbitrarily fixed symbol such that  $\star \notin M_G$ , called the *enabling relation* that satisfies:
  - ▷ (E1) If  $\star \vdash_G m$ , then  $\lambda_G(m) = (O, Q, 0)$  and  $n = \star$  whenever  $n \vdash_G m$
  - ▷ (E2) If  $m \vdash_G n$  and  $\lambda_G^{\text{QA}}(n) = A$ , then  $\lambda_G^{\text{QA}}(m) = Q$  and  $\lambda_G^{\mathbb{N}}(m) = \lambda_G^{\mathbb{N}}(n)$
  - ▷ (E3) If  $m \vdash_G n$  and  $m \neq \star$ , then  $\lambda_G^{\text{OP}}(m) \neq \lambda_G^{\text{OP}}(n)$
  - ▷ (E4) If  $m \vdash_G n$ ,  $m \neq \star$  and  $\lambda_G^{\mathbb{N}}(m) \neq \lambda_G^{\mathbb{N}}(n)$ , then  $\lambda_G^{\text{OP}}(m) = O$  (and  $\lambda_G^{\text{OP}}(n) = P$ )

in which  $\lambda_G^{\text{OP}} \stackrel{\text{df.}}{=} \lambda_G; \pi_1 : M_G \rightarrow \{O, P\}$ ,  $\lambda_G^{\text{QA}} \stackrel{\text{df.}}{=} \lambda_G; \pi_2 : M_G \rightarrow \{Q, A\}$ ,  $\lambda_G^{\mathbb{N}} \stackrel{\text{df.}}{=} \lambda_G; \pi_3 : M_G \rightarrow \mathbb{N}$ . A move  $m \in M_G$  is *initial* if  $\star \vdash_G m$ , an *O-move* (resp. a *P-move*) if  $\lambda_G^{\text{OP}}(m) = O$  (resp. if  $\lambda_G^{\text{OP}}(m) = P$ ), a *question* (resp. an *answer*) if  $\lambda_G^{\text{QA}}(m) = Q$  (resp. if  $\lambda_G^{\text{QA}}(m) = A$ ), and *internal* (resp. *external*) if  $\lambda_G^{\mathbb{N}}(m) > 0$  (resp. if  $\lambda_G^{\mathbb{N}}(m) = 0$ ). A sequence  $s \in M_G^*$  is called *d-complete* ( $d \in \mathbb{N} \cup \{\omega\}$ ) if it ends with an external or  $d'$ -internal move with  $d' > d$ , where  $\omega$  is the *least transfinite ordinal*. We write  $M_G^{\text{init}}$  for the set of all initial moves in  $G$ .

That is, our variant of arena is an arena in [AM99] equipped with the *degree of internality*  $\lambda_G^{\mathbb{N}}$  on moves<sup>1</sup> and satisfying some additional axioms:

- The set  $\{\pi_1(m) \mid m \in M_G\}$  is required to be finite, so that each move is distinguishable.
- The condition on the labeling function requires an upper bound of degrees of internality.
- E1 adds  $\lambda_G^{\mathbb{N}}(m) = 0$  if  $m \in M_G$  is initial as Opponent cannot “see” internal moves.
- E2 additionally requires the degree of internality between a “QA-pair” to be the same.
- E4 determines that only Player can make a move for a previous move if they have different degrees of internality because internal moves are “invisible” to Opponent.

From now on, the word *arenas* refers to the variant defined above. Given an arena, we are interested in certain finite sequences of its moves equipped with a *justifying* relation:

► **Definition 2.2.2** (Justified sequences [HO00, AM99, McC98]). A *justified sequence* (*j-sequence*) in an arena  $G$  is a finite sequence  $s \in M_G^*$ , in which each non-initial move  $n$  is associated with (or *points* at) a unique move  $m$ , called the *justifier* of  $n$  in  $s$ , that occurs previously in  $s$  and satisfies  $m \vdash_G n$ . We say that  $n$  is *justified* by  $m$ , or there is a *pointer* from  $n$  to  $m$ .

<sup>1</sup>We need all natural numbers for  $\lambda_G^{\mathbb{N}}$ , not only the *internal/external* (I/E) distinction, to define a *step-by-step* execution of the *hiding operation* (see [YA16] for the details).

► **Notation.** We write  $\mathcal{J}_s(n)$  for the justifier of a non-initial move  $n$  in a j-sequence  $s$ , where  $\mathcal{J}_s$  is the “function of pointers in  $s$ ”, and  $\mathcal{J}_G$  for the set of all j-sequences in an arena  $G$ .

The idea is that each non-initial move in a j-sequence must be made for a specific previous move, called its *justifier*. Note that the first element  $m$  of each non-empty j-sequence  $ms \in \mathcal{J}_G$  is an initial move in  $G$ ; we call  $m$  the *opening move* of  $ms$  and write  $\mathcal{O}(ms)$  for it.

We may consider justifiers from the “external viewpoint”:

► **Definition 2.2.3** (External justifiers [YA16]). Let  $G$  be an arena, and  $s \in \mathcal{J}_G$ ,  $d \in \mathbb{N} \cup \{\omega\}$ . Each non-initial move  $n$  in  $s$  has a unique sequence of justifiers  $nm_1m_2 \dots m_km$  ( $k \geq 0$ ), i.e.,  $\mathcal{J}_s(n) = m_1$ ,  $\mathcal{J}_s(m_1) = m_2$ ,  $\dots$ ,  $\mathcal{J}_s(m_{k-1}) = m_k$ ,  $\mathcal{J}_s(m_k) = m$ , such that  $m_1, m_2, \dots, m_k$  are  $d'$ -internal with  $0 < d' \leq d$  but  $m$  is not. We call  $m$  the  *$d$ -external justifier* of  $n$  in  $s$ .

► **Notation.** We usually write  $\mathcal{J}_s^{\odot d}(n)$  for the  $d$ -external justifier of  $n$  in a j-sequence  $s$ .

► **Definition 2.2.4** (External justified subsequences [YA16]). Let  $s$  be a j-sequence in an arena  $G$  and  $d \in \mathbb{N} \cup \{\omega\}$ . The  *$d$ -external justified (j-) subsequence*  $\mathcal{H}_G^d(s)$  of  $s$  is obtained from  $s$  by deleting  $d'$ -internal moves,  $0 < d' \leq d$ , equipped with the pointers  $\mathcal{J}_s^{\odot d}$ .

► **Definition 2.2.5** (Hiding operation on arenas [YA16]). Let  $d \in \mathbb{N} \cup \{\omega\}$ , and  $G$  an arena. The arena  $\mathcal{H}^d(G)$  is defined by  $M_{\mathcal{H}^d(G)} \stackrel{\text{df.}}{=} \{m \in M_G \mid \lambda_G^{\mathbb{N}}(m) = 0 \vee \lambda_G^{\mathbb{N}}(m) > d\}$ ,  $\lambda_{\mathcal{H}^d(G)} \stackrel{\text{df.}}{=} \lambda_G^{\odot d} \upharpoonright M_{\mathcal{H}^d(G)}$ ,  $\lambda_G^{\odot d} \stackrel{\text{df.}}{=} \langle \lambda_G^{\text{OP}}, \lambda_G^{\text{QA}}, n \mapsto \lambda_G^{\mathbb{N}}(n) \odot d, x \odot d \stackrel{\text{df.}}{=} \begin{cases} x - d & \text{if } x \geq d \\ 0 & \text{otherwise} \end{cases} \text{ for all } x \in \mathbb{N}, \text{ and } m \vdash_{\mathcal{H}^d(G)} n \stackrel{\text{df.}}{\iff} \exists k \in \mathbb{N}, m_1, m_2, \dots, m_{2k-1}, m_{2k} \in M_G \setminus M_{\mathcal{H}^d(G)}. m \vdash_G m_1 \wedge m_1 \vdash_G m_2 \wedge \dots \wedge m_{2k-1} \vdash_G m_{2k} \wedge m_{2k} \vdash_G n \text{ (note that } m \vdash_G n \text{ if } k = 0) \rangle$ .

I.e.,  $\mathcal{H}^d(G)$  is obtained from  $G$  by deleting all  $d'$ -internal moves,  $0 < d' \leq d$ , decreasing by  $d$  the degree of internality of the remaining moves and “concatenating” the enabling relation to form the “ $d$ -external” one. We clearly have:

► **Lemma 2.2.6** (Closure of arenas and j-sequences under hiding [YA16]). If  $G$  is an arena, then so is  $\mathcal{H}^d(G)$  for all  $d \in \mathbb{N} \cup \{\omega\}$  such that  $\mathcal{H}^0(G) = G$  and  $\mathcal{H}_G^d(s) \in \mathcal{J}_{\mathcal{H}^d(G)}$  for all  $s \in \mathcal{J}_G$ .

Next, let us recall the notion of “relevant part” of previous moves, called *views*:

► **Definition 2.2.7** (Views [HO00, AM99, McC98]). Given a j-sequence  $s$  in an arena  $G$ , we define the *Player view (P-view)*  $\lceil s \rceil_G$  and the *Opponent view (O-view)*  $\lfloor s \rfloor_G$  by induction on the length of  $s$  as follows:

- $\lceil \epsilon \rceil_G \stackrel{\text{df.}}{=} \epsilon$
- $\lceil sm \rceil_G \stackrel{\text{df.}}{=} \lceil s \rceil_G.m$  if  $m$  is a P-move
- $\lceil sm \rceil_G \stackrel{\text{df.}}{=} m$  if  $m$  is initial
- $\lceil smtn \rceil_G \stackrel{\text{df.}}{=} \lceil s \rceil_G.mn$  if  $n$  is an O-move with  $\mathcal{J}_{smtn}(n) = m$
- $\lfloor \epsilon \rfloor_G \stackrel{\text{df.}}{=} \epsilon$
- $\lfloor sm \rfloor_G \stackrel{\text{df.}}{=} \lfloor s \rfloor_G.m$  if  $m$  is an O-move
- $\lfloor smtn \rfloor_G \stackrel{\text{df.}}{=} \lfloor s \rfloor_G.mn$  if  $n$  is a P-move with  $\mathcal{J}_{smtn}(n) = m$

where the justifiers of the remaining moves in  $\lceil s \rceil_G$  (resp.  $\lfloor s \rfloor_G$ ) are unchanged if they occur in  $\lceil s \rceil_G$  (resp.  $\lfloor s \rfloor_G$ ) and undefined otherwise.

► **Notation.** We omit the subscript  $G$  in  $\lceil s \rceil_G$ ,  $\lfloor s \rfloor_G$  when the underlying game  $G$  is obvious.

The idea behind this definition is as follows. Given a “position” or prefix  $tm$  of a  $j$ -sequence  $s$  in an arena  $G$  such that  $m$  is a P-move (resp. an O-move), the P-view  $\lceil t \rceil$  (resp. the O-view  $\lfloor t \rfloor$ ) is intended to be the currently “relevant” part of  $t$  for Player (resp. Opponent). That is, Player (resp. Opponent) is concerned only with the last O-move (resp. P-move), its justifier and that justifier’s “concern”, i.e., P-view (resp. O-view), which then recursively proceeds.

We are now ready to define:

► **Definition 2.2.8** (Legal positions [YA16]). A *(dynamic) legal position* in an arena  $G$  is a sequence  $s \in M_G^*$  (equipped with justifiers) that satisfies:

- **Justification.**  $s$  is a  $j$ -sequence in  $G$ .
  - **Alternation.** If  $s = s_1 m n s_2$ , then  $\lambda_G^{\text{OP}}(m) \neq \lambda_G^{\text{OP}}(n)$ .
  - **Generalized visibility.** If  $s = t m u$  with  $m$  non-initial and  $d \in \mathbb{N} \cup \{\omega\}$  satisfy  $\lambda_G^{\mathbb{N}}(m) = 0 \vee \lambda_G^{\mathbb{N}}(m) > d$ , then  $\mathcal{J}_s^{\ominus d}(m)$  occurs in  $\lceil \mathcal{H}_G^d(t) \rceil_{\mathcal{H}^d(G)}$  if  $m$  is a P-move, and it occurs in  $\lfloor \mathcal{H}_G^d(t) \rfloor_{\mathcal{H}^d(G)}$  if  $m$  is an O-move.
  - **IE-switch.** If  $s = s_1 m n s_2$  with  $\lambda_G^{\mathbb{N}}(m) \neq \lambda_G^{\mathbb{N}}(n)$ , then  $m$  is an O-move.
- **Notation.** We write  $L_G$  for the set of all legal positions in an arena  $G$ .
- I.e., our (dynamic) legal positions are legal positions in [AM99] satisfying additional axioms:
- Generalized visibility is a natural generalization of *visibility* [HO00, AM99, McC98]; it requires that visibility holds after any iteration of the “*hiding operation on arenas*” [YA16].
  - IE-switch states that only Player can change the degree of internality during a play because internal moves are “invisible” to Opponent.

From now on, the word *legal positions* refers to the variant defined above by default.

Next, note that in a legal position in an arena, there may be several initial moves; the legal position consists of *chains of justifiers* initiated by such initial moves, and chains with the same initial move form a *thread*. Formally,

► **Definition 2.2.9** (Threads [AM99, McC98]). Let  $G$  be an arena, and  $s \in L_G$ . Assume that  $m$  is an occurrence of a move in  $s$ . The *chain of justifiers* from  $m$  is a sequence  $m_0 m_1 \dots m_k \in M_G^*$  such that  $k \geq 0$ ,  $m_k = m$ ,  $\mathcal{J}_s(m_k) = m_{k-1}$ ,  $\mathcal{J}_s(m_{k-1}) = m_{k-2}$ ,  $\dots$ ,  $\mathcal{J}_s(m_1) = m_0$ , and  $m_0$  is initial. In this case, we say that  $m$  is *hereditarily justified* by  $m_0$ . The subsequence of  $s$  consisting of the chains of justifiers in which  $m_0$  occurs is called the *thread* of  $m_0$  in  $s$ . An occurrence of an initial move is often called an *initial occurrence*.

► **Notation.** We write  $s \upharpoonright I$ , where  $s \in L_G$  and  $I$  is a set of initial occurrences in  $s$ , for the subsequence of  $s$  consisting of threads of initial occurrences in  $I$ , and define  $s \upharpoonright m \stackrel{\text{df.}}{=} s \upharpoonright \{m\}$ .

We are now ready to define our variant of games:

► **Definition 2.2.10** (Games). A *(dynamic) game* is a tuple  $G = (M_G, \lambda_G, \vdash_G, P_G, \simeq_G)$ , where:

- The triple  $(M_G, \lambda_G, \vdash_G)$  forms an arena

- $P_G$  is a subset of  $L_G$  whose elements are called *(valid) positions* in  $G$  that satisfies:
  - ▷ (V1)  $P_G$  is non-empty and *prefix-closed* (i.e.,  $sm \in P_G \Rightarrow s \in P_G$ )
  - ▷ (V2) If  $s \in P_G$  and  $I$  is a set of initial occurrences in  $s$ , then  $s \upharpoonright I \in P_G$
  - ▷ (V3) For any  $sm, s'm' \in P_G^{\text{odd}}$ ,  $i \in \mathbb{N}$  such that  $i < \lambda_G^{\mathbb{N}}(m) = \lambda_G^{\mathbb{N}}(m')$ , if  $\mathcal{H}_G^i(s) = \mathcal{H}_G^i(s')$ , then  $m = m'$  and  $\mathcal{J}_{sm}^{\odot i}(m) = \mathcal{J}_{s'm'}^{\odot i}(m')$
- $\simeq_G$  is an equivalence relation on  $P_G$  called the *identification of positions* that satisfies:
  - ▷ (I1)  $s \simeq_G t \Rightarrow \pi_1^*(s) = \pi_1^*(t)$
  - ▷ (I2)  $sm \simeq_G tn \Rightarrow s \simeq_G t \wedge \lambda_G(m) = \lambda_G(n) \wedge (m, n \in M_G^{\text{init}} \vee \exists i \in \mathbb{N}^+. \mathcal{J}_{sm}(m) = s_i \wedge \mathcal{J}_{tn}(n) = t_i)$
  - ▷ (I3)  $\forall d \in \mathbb{N} \cup \{\omega\}. s \simeq_G^d t \wedge sm \in P_G \Rightarrow \exists tn \in P_G. sm \simeq_G^d tn$ , where  $u \simeq_G^d v \stackrel{\text{df.}}{\Leftrightarrow} \exists u', v' \in P_G. u' \simeq_G v' \wedge \mathcal{H}_G^d(u') = \mathcal{H}_G^d(u) \wedge \mathcal{H}_G^d(v') = \mathcal{H}_G^d(v)$  for all  $u, v \in P_G$ .

I.e., our variant of games is dynamic games [YA16] equipped with *identification of positions* that is to “ignore” permutations of tags in exponential ! as in [AJM00, McC98].

► **Definition 2.2.11** (Finitely well-opened games). A game  $G$  is *finitely well-opened* if  $[m]_e \in M_G^{\text{init}}$  implies  $e = \epsilon$ , and  $s.[m] \in P_G$  with  $[m]$  initial implies  $s = \epsilon$ .

I.e., a game is finitely well-opened if it is *well-opened* [AM99, McC98] and its initial moves have the empty tag  $\epsilon$  only. From now on, *games* refer to *finitely well-opened (dynamic) games*.

- Example 2.2.12. The *terminal game*  $I$  is defined by  $I \stackrel{\text{df.}}{=} (\emptyset, \emptyset, \emptyset, \{\epsilon\}, \{(\epsilon, \epsilon)\})$ .
- Example 2.2.13. The *boolean game*  $\mathbf{2}$  is defined by:

- $M_{\mathbf{2}} \stackrel{\text{df.}}{=} \{q, \top, \perp\}$ , where each move has the empty tag  $\epsilon$
- $\lambda_{\mathbf{2}} : q \mapsto (O, Q, 0), \top \mapsto (P, A, 0), \perp \mapsto (P, A, 0)$
- $\vdash_{\mathbf{2}} \stackrel{\text{df.}}{=} \{(\star, q), (q, \top), (q, \perp)\}$
- $P_{\mathbf{2}} \stackrel{\text{df.}}{=} \text{pref}(\{q.\top, q.\perp\})$ , where each non-initial move is justified by  $q$
- $s \simeq_{\mathbf{2}} t \stackrel{\text{df.}}{\Leftrightarrow} s = t$ .

The positions  $q.\top, q.\perp$  are intended to represent *true* and *false*, respectively.

- Example 2.2.14. The *natural number game*  $N$  is defined by:

- $M_N \stackrel{\text{df.}}{=} \{\hat{q}, q, \bullet, b\}$ , where each move has the tag  $\epsilon$ , and we often abbreviate  $\hat{q}$  as  $q$
- $\lambda_N : \hat{q} \mapsto (O, Q, 0), q \mapsto (O, Q, 0), \bullet \mapsto (P, A, 0), b \mapsto (P, A, 0)$
- $\vdash_N \stackrel{\text{df.}}{=} \{(\star, \hat{q}), (\hat{q}, b), (\hat{q}, \bullet), (q, b), (q, \bullet), (\bullet, q)\}$
- $P_N \stackrel{\text{df.}}{=} \text{pref}(\{\hat{q}.(\bullet.q)^n.b \mid n \in \mathbb{N}\})$ , where each non-initial move is justified by the last move
- $s \simeq_N t \stackrel{\text{df.}}{\Leftrightarrow} s = t$ .

The position  $\hat{q}.(\bullet.q)^n.b$  is to represent  $n \in \mathbb{N}$ . Let us define  $\underline{n} \stackrel{\text{df.}}{=} \text{pref}(\{\hat{q}.(\bullet.q)^n.b\})^{\text{even}}$ .



► Example 2.2.15. The *tag game*  $\mathcal{G}(\mathcal{T})$  is defined by:

- $M_{\mathcal{G}(\mathcal{T})} \stackrel{\text{df.}}{=} \{\hat{q}_{\mathcal{T}}, q_{\mathcal{T}}, \sharp, |, \checkmark, \langle, \rangle\}$ , where each move has the tag  $\epsilon$ , and  $\hat{q}_{\mathcal{T}}$  is often written  $q_{\mathcal{T}}$
- $\lambda_{\mathcal{G}(\mathcal{T})} : \hat{q}_{\mathcal{T}} \mapsto (O, Q, 0), q_{\mathcal{T}} \mapsto (O, Q, 0), e \mapsto (P, A, 0)$ , where  $e \in \{\sharp, |, \checkmark, \langle, \rangle\}$
- $\vdash_{\mathcal{G}(\mathcal{T})} \stackrel{\text{df.}}{=} \{(\star, \hat{q}_{\mathcal{T}})\} \cup \{(x, y) \mid x \in \{\hat{q}_{\mathcal{T}}, q_{\mathcal{T}}\}, y \in \{\sharp, |, \checkmark, \langle, \rangle\}\} \cup \{(e, q_{\mathcal{T}}) \mid e \in \{\sharp, |, \checkmark, \langle, \rangle\}\}$
- $P_{\mathcal{G}(\mathcal{T})} \stackrel{\text{df.}}{=} \text{pref}(\{\hat{q}_{\mathcal{T}}e_1q_{\mathcal{T}}e_2 \dots q_{\mathcal{T}}e_kq_{\mathcal{T}}\checkmark \mid k \in \mathbb{N}, e_1e_2 \dots e_k \in \mathcal{T}\})$ , where each non-initial move is justified by the last move
- $s \simeq_{\mathcal{G}(\mathcal{T})} t \stackrel{\text{df.}}{\iff} s = t$ .

The position  $\hat{q}_{\mathcal{T}}e_1q_{\mathcal{T}}e_2 \dots q_{\mathcal{T}}e_kq_{\mathcal{T}}\checkmark$  is intended to represent the tag  $e_1e_2 \dots e_k \in \mathcal{T}$ .

► **Definition 2.2.16** (Subgames). A *subgame* of a game  $G$  is a game  $H$  that satisfies  $M_H \subseteq M_G$ ,  $\lambda_H = \lambda_G \upharpoonright M_H$ ,  $\vdash_H \subseteq \vdash_G \cap (\{\star\} \cup M_H) \times M_H$ ,  $P_H \subseteq P_G$ ,  $\simeq_H \subseteq \simeq_G$ . In this case, we write  $H \trianglelefteq G$ .

► **Definition 2.2.17** (Hiding operation on games [YA16]). The *d-hiding operation*  $\mathcal{H}^d$  on games for each  $d \in \mathbb{N} \cup \{\omega\}$  is defined as follows. Given a game  $G$ ,  $\mathcal{H}^d(G)$  is the game such that  $(M_{\mathcal{H}^d(G)}, \lambda_{\mathcal{H}^d(G)}, \vdash_{\mathcal{H}^d(G)})$  is the arena  $\mathcal{H}^d(G)$ ,  $P_{\mathcal{H}^d(G)} \stackrel{\text{df.}}{=} \{\mathcal{H}_G^d(s) \mid s \in P_G\}$ , and  $\mathcal{H}_G^d(s) \simeq_{\mathcal{H}^d(G)}$   $\mathcal{H}_G^d(t) \stackrel{\text{df.}}{\iff} s \simeq_G^d t$ . A game  $G$  is *static* if  $\mathcal{H}^\omega(G) = G$ .

► **Theorem 2.2.18** (Closure of games under hiding). For any game  $G$ ,  $\mathcal{H}^d(G)$  forms a well-defined game for all  $d \in \mathbb{N} \cup \{\omega\}$ . Moreover, if  $H \trianglelefteq G$ , then  $\mathcal{H}^d(H) \trianglelefteq \mathcal{H}^d(G)$  for all  $d \in \mathbb{N} \cup \{\omega\}$ .

*Proof.* First,  $\simeq_{\mathcal{H}^d(G)}$  is well-defined as  $\mathcal{H}_G^d(s) \simeq_{\mathcal{H}^d(G)} \mathcal{H}_G^d(t)$  does not depend on the representatives  $s, t \in P_G$ . By the corresponding result in [YA16], it suffices to verify the preservation of the axioms l1, l2, l3 under  $\mathcal{H}^d$ . Then l1, l2 for  $\mathcal{H}^d(G)$  immediately follow from l2 on  $G$ . For l3, if  $\mathcal{H}_G^d(s) \simeq_{\mathcal{H}^d(G)}^{d'} \mathcal{H}_G^d(t)$  and  $\mathcal{H}_G^d(s).m \in P_{\mathcal{H}^d(G)}$ , where we assume  $d \neq \omega \wedge d' \neq \omega$  since otherwise l3 on  $\mathcal{H}^d(G)$  is reduced to that on  $G$ , then  $\exists s'm \in P_G. \mathcal{H}_G^d(s'm) = \mathcal{H}_G^d(s).m$  and  $\mathcal{H}_G^{d+d'}(s') = \mathcal{H}_G^{d+d'}(s) \simeq_{\mathcal{H}^{d+d'}(G)} \mathcal{H}_G^{d+d'}(t)$  (see [YA16] for the proof); thus by l3 on  $G$ , we may conclude  $\exists tn \in P_G. s'm \simeq_G^{d+d'} tn$ , whence  $\exists \mathcal{H}_G^d(t).n \in P_{\mathcal{H}^d(G)}. \mathcal{H}_G^d(s).m = \mathcal{H}_G^d(s'm) \simeq_{\mathcal{H}^d(G)}^{d'} \mathcal{H}_G^d(tn) = \mathcal{H}_G^d(t).n$ . Finally for  $H \trianglelefteq G \Rightarrow \mathcal{H}^d(H) \trianglelefteq \mathcal{H}^d(G)$ , again by the result in [YA16] it suffices to show  $\simeq_H \subseteq \simeq_G \Rightarrow \simeq_{\mathcal{H}^d(H)} \subseteq \simeq_{\mathcal{H}^d(G)}$ , but it is immediate from the definition. ■

At the end of the present section, we define constructions on games based on the standard ones [McC98, AM99, YA16] with tags explicit, equipping them with constructions on identification of positions. For this, like *variable convention* [Han94], we assume that there are countably infinite copies of the symbols  $|, \sharp, \langle, \rangle$ , and write  $|\alpha, \sharp_\beta$ , etc., where  $\alpha, \beta \in \{0, 1\}^*$ , for these copies. However, for readability, we usually omit these subscripts  $\alpha$  unless necessary. Also, we define  $e_\alpha \stackrel{\text{df.}}{=} (e_1)_\alpha (e_2)_\alpha \dots (e_k)_\alpha$  for all  $e = e_1e_2 \dots e_k \in \mathcal{T}$ ,  $\alpha \in \{0, 1\}^*$ .

► **Definition 2.2.19** (Tensor [AM99, McC98]). The *tensor (product)*  $A \otimes B$  of games  $A, B$  is defined by:

- $M_{A \otimes B} \stackrel{\text{df.}}{=} \{[(m, 0)]_{e_0} \mid [m]_e \in M_A\} \cup \{[(m', 1)]_{e'_1} \mid [m']_{e'} \in M_B\}$
- $\lambda_{A \otimes B}([(m, i)]_{e_i}) \stackrel{\text{df.}}{=} \begin{cases} \lambda_A([m]_e) & \text{if } i = 0 \\ \lambda_B([m]_{e'}) & \text{if } i = 1 \end{cases}$

- $\star \vdash_{A \otimes B} [(m, i)]_{e_i} \stackrel{\text{df.}}{\iff} (i = 0 \wedge \star \vdash_A [m]_e) \vee (i = 1 \wedge \star \vdash_B [m]_e)$
- $[(m, i)]_{e_i} \vdash_{A \otimes B} [(m', j)]_{e'_j} \stackrel{\text{df.}}{\iff} (i = 0 = j \wedge [m]_e \vdash_A [m']_{e'}) \vee (i = 1 = j \wedge [m]_e \vdash_B [m']_{e'})$
- $P_{A \otimes B} \stackrel{\text{df.}}{=} \{s \in L_{A \otimes B} \mid s \upharpoonright 0 \in P_A, s \upharpoonright 1 \in P_B\}$ , where  $s \upharpoonright i$  is the subsequence of  $s$  with the justifiers in  $s$  that consists of moves  $[(m, i)]_{e_i}$  changed into  $[m]_e$
- $s \simeq_{A \otimes B} t \stackrel{\text{df.}}{\iff} (\pi_2 \circ \pi_1)^*(s) = (\pi_2 \circ \pi_1)^*(t) \wedge s \upharpoonright 0 \simeq_A t \upharpoonright 0 \wedge s \upharpoonright 1 \simeq_B t \upharpoonright 1$ .

► **Definition 2.2.20** (Linear implication [AM99, McC98]). The *linear implication*  $A \multimap B$  from a static game  $A$  to another game  $B$  is defined by:

- $M_{A \multimap B} \stackrel{\text{df.}}{=} \{[(m, 0)]_{e_0} \mid [m]_e \in M_A\} \cup \{[(m', 1)]_{e'_1} \mid [m']_{e'} \in M_B\}$
- $\lambda_{A \multimap B}([(m, i)]_{e_i}) \stackrel{\text{df.}}{=} \begin{cases} \overline{\lambda_A}([m]_e) & \text{if } i = 0 \\ \lambda_B([m]_e) & \text{if } i = 1 \end{cases}, \overline{\lambda_A} \stackrel{\text{df.}}{=} \langle \overline{\lambda_A^{\text{OP}}}, \lambda_A^{\text{QA}}, \lambda_A^{\text{N}} \rangle, \overline{\lambda_A^{\text{OP}}}(x) \stackrel{\text{df.}}{=} \begin{cases} P & \text{if } \lambda_A^{\text{OP}}(x) = O \\ O & \text{otherwise} \end{cases}$
- $\star \vdash_{A \multimap B} [(m, i)]_{e_i} \stackrel{\text{df.}}{\iff} i = 1 \wedge \star \vdash_B [m]_e$
- $[(m, i)]_{e_i} \vdash_{A \multimap B} [(m', j)]_{e'_j} \stackrel{\text{df.}}{\iff} (i = 0 = j \wedge [m]_e \vdash_A [m']_{e'}) \vee (i = 1 = j \wedge [m]_e \vdash_B [m']_{e'}) \vee (i = 1 \wedge j = 0 \wedge \star \vdash_B [m]_e \wedge \star \vdash_A [m']_{e'})$
- $P_{A \multimap B} \stackrel{\text{df.}}{=} \{s \in L_{A \multimap B} \mid s \upharpoonright 0 \in P_A, s \upharpoonright 1 \in P_B\}$
- $s \simeq_{A \multimap B} t \stackrel{\text{df.}}{\iff} (\pi_2 \circ \pi_1)^*(s) = (\pi_2 \circ \pi_1)^*(t) \wedge s \upharpoonright 0 \simeq_A t \upharpoonright 0 \wedge s \upharpoonright 1 \simeq_B t \upharpoonright 1$ .

► **Definition 2.2.21** (Product [AM99, McC98]). The *product*  $A \& B$  of games  $A, B$  is defined by:

- $M_{A \& B} \stackrel{\text{df.}}{=} \{[(m, 0)]_{e_0} \mid [m]_e \in M_A\} \cup \{[(m', 1)]_{e'_1} \mid [m']_{e'} \in M_B\}$
- $\lambda_{A \& B}([(m, i)]_{e_i}) \stackrel{\text{df.}}{=} \begin{cases} \lambda_A([m]_e) & \text{if } i = 0 \\ \lambda_B([m]_e) & \text{if } i = 1 \end{cases}$
- $\star \vdash_{A \& B} [(m, i)]_{e_i} \stackrel{\text{df.}}{\iff} (i = 0 \wedge \star \vdash_A [m]_e) \vee (i = 1 \wedge \star \vdash_B [m]_e)$
- $[(m, i)]_{e_i} \vdash_{A \& B} [(m', j)]_{e'_j} \stackrel{\text{df.}}{\iff} (i = 0 = j \wedge [m]_e \vdash_A [m']_{e'}) \vee (i = 1 = j \wedge [m]_e \vdash_B [m']_{e'})$
- $P_{A \& B} \stackrel{\text{df.}}{=} \{s \in L_{A \& B} \mid s \upharpoonright 0 \in P_A, s \upharpoonright 1 \in P_B\}$
- $s \simeq_{A \& B} t \stackrel{\text{df.}}{\iff} (\pi_2 \circ \pi_1)^*(s) = (\pi_2 \circ \pi_1)^*(t) \wedge s \upharpoonright 0 \simeq_A t \upharpoonright 0 \wedge s \upharpoonright 1 \simeq_B t \upharpoonright 1$ .

► **Definition 2.2.22** (Generalized product [YA16]). The *generalized product*  $L \& R$  of games  $L, R$  such that  $\mathcal{H}^\omega(L) \trianglelefteq C \multimap A, \mathcal{H}^\omega(R) \trianglelefteq C \multimap B$  for some static games  $A, B, C$  is defined by:

- $M_{L \& R} \stackrel{\text{df.}}{=} \{[(m, 0)]_{e_0} \mid [m]_e \in M_{(C, 0)} \cap (M_L \cup M_R)\} \cup \{[(m, 0)]_{e_0} \mid [m]_e \in M_L \setminus M_{(C, 0)}\} \cup \{[(m', 1)]_{e'_1} \mid [m']_{e'} \in M_R \setminus M_{(C, 0)}\}$ , where  $M_{(C, 0)} \stackrel{\text{df.}}{=} \{[(c, 0)]_{e_0} \mid [c]_e \in M_C\}$
- $\lambda_{L \& R}([(m, i)]_{e_i}) \stackrel{\text{df.}}{=} \begin{cases} \lambda_C([c]_{e'}) & \text{if } [(m, i)]_{e_i} = [(c, 0), 0]_{e'_{00}} \text{ and } [(c, 0)]_{e'_0} \in M_{(C, 0)} \\ \lambda_L([m]_e) & \text{if } i = 0 \text{ and } [m]_e \notin M_{(C, 0)} \\ \lambda_R([m]_e) & \text{if } i = 1 \end{cases}$

- $\star \vdash_{L\&R} [(m, i)]_{e_i} \stackrel{\text{df.}}{\Leftrightarrow} (i = 0 \wedge \star \vdash_L [m]_e) \vee (i = 1 \wedge \star \vdash_R [m]_e)$
- $[(m, i)]_{e_i} \vdash_{L\&R} [(m', j)]_{e'_j} \stackrel{\text{df.}}{\Leftrightarrow} (i = 0 = j \wedge [m]_e \vdash_L [m']_{e'}) \vee (i = 1 = j \wedge [m]_e \vdash_R [m']_{e'}) \vee ([m]_e, [m']_{e'} \in M_{(C, 0)} \wedge [m]_e \vdash_R [m']_{e'}) \vee (i = 1 \wedge j = 0 \wedge [m]_e \vdash_R [m']_{e'})$
- $P_{L\&R} \stackrel{\text{df.}}{=} \{s \in L_{L\&R} \mid s \upharpoonright L \in P_L, s \upharpoonright R = \epsilon\} \cup \{t \in L_{L\&R} \mid t \upharpoonright L = \epsilon, t \upharpoonright R \in P_R\}$ , where  $s \upharpoonright L$  (resp.  $s \upharpoonright R$ ) is the subsequence of  $s$  with the justifiers in  $s$  that consists of moves  $[(m, i)]_{e_i}$  hereditarily justified by an opening move  $[((a, 1), 0)]$  with  $[a] \in M_A^{\text{init}}$  (resp.  $[((b, 1), 1)]$  with  $[b] \in M_B^{\text{init}}$ ) changed into  $[m]_e$
- $s \simeq_{L\&R} t \stackrel{\text{df.}}{\Leftrightarrow} (\pi_2 \circ \pi_1)^*(s) = (\pi_2 \circ \pi_1)^*(t) \wedge s \upharpoonright L \simeq_L t \upharpoonright L \wedge s \upharpoonright R \simeq_R t \upharpoonright R$ .
- **Definition 2.2.23** (Exponential [McC98]). The *exponential*  $!A$  of a game  $A$  is defined by:
  - $M_{!A} \stackrel{\text{df.}}{=} \{[m]_{\langle f \rangle \# e} \mid [m]_e \in M_A, f \in \mathcal{T}\}$  and  $\lambda_{!A}([m]_{\langle f \rangle \# e}) \stackrel{\text{df.}}{=} \lambda_A([m]_e)$
  - $\star \vdash_{!A} [m]_{\langle f \rangle \# e} \stackrel{\text{df.}}{\Leftrightarrow} \star \vdash_A [m]_e$  and  $[m]_{\langle f \rangle \# e} \vdash_{!A} [m']_{\langle f' \rangle \# e'} \stackrel{\text{df.}}{\Leftrightarrow} f = f' \wedge [m]_e \vdash_A [m']_{e'}$
  - $P_{!A} \stackrel{\text{df.}}{=} \{s \in L_{!A} \mid \forall i \in \mathbb{N}. s \upharpoonright i \in P_A\}$ , where  $s \upharpoonright i$  is the subsequence of  $s$  with the justifiers in  $s$  consisting of moves  $[m]_{\langle f \rangle \# e}$  such that  $\text{ede}(\langle f \rangle) = i$  but changed into  $[m]_e$
  - $s \simeq_{!A} t \stackrel{\text{df.}}{\Leftrightarrow} \exists \varphi \in \mathcal{P}(\mathbb{N}). (\pi_1 \circ \text{ede} \circ \pi_2)^*(s) = (\varphi \circ \pi_1 \circ \text{ede} \circ \pi_2)^*(t) \wedge \forall i \in \mathbb{N}. s \upharpoonright \varphi(i) \simeq_A t \upharpoonright i$ , where  $\mathcal{P}(\mathbb{N})$  denotes the set of all permutations of natural numbers.

I.e., our exponential  $!A$  is a slight modification of the one in [McC98] by generalizing threads  $[m]_{i\#e}$  to  $[m]_{\langle f \rangle \# e}$  ( $[m]_e \in M_A$ ). Since we are focusing on *well-opened* games  $A$ , there is at most one thread with a tag  $\langle f \rangle \# e$  such that  $\text{ede}(\langle f \rangle) = i$  for each  $i \in \mathbb{N}$  in a position of  $!A$ . As a consequence, our exponential  $!A$  is the same as the one in [McC98] except that there is a choice in the implementation  $\langle f \rangle$  of tags  $i \in \mathbb{N}$  (but that implementation  $\langle f \rangle$  is unique within  $!A$ ).

► **Definition 2.2.24** (Concatenation [YA16]). Let  $J, K$  be games such that  $\mathcal{H}^\omega(J) \trianglelefteq A \multimap B$ ,  $\mathcal{H}^\omega(K) \trianglelefteq B \multimap C$  for some static games  $A, B, C$ . Their *concatenation*  $J \ddagger K$  is defined by:

- $M_{J \ddagger K} \stackrel{\text{df.}}{=} \{[(m, 0)]_{e_0} \mid [m]_e \in M_J\} \cup \{[(m', 1)]_{e'_1} \mid [m']_{e'} \in M_K\}$
- $\lambda_{J \ddagger K}([(m, i)]_{e_i}) \stackrel{\text{df.}}{=} \begin{cases} \lambda_J^{+\mu}([m]_e) & \text{if } i = 0 \text{ and } [m]_e \text{ comes from } B \\ \lambda_J([m]_e) & \text{if } i = 0 \text{ and } [m]_e \text{ does not come from } B \\ \lambda_K^{+\mu}([m]_e) & \text{if } i = 1 \text{ and } [m]_e \text{ comes from } B \\ \lambda_K([m]_e) & \text{if } i = 1 \text{ and } [m]_e \text{ does not come from } B \end{cases}$ , where  $\mu \in \mathbb{N}^+$
- is defined by  $\mu \stackrel{\text{df.}}{=} \sup(\{\lambda_J^{\mathbb{N}}([m]_e) \mid [m]_e \in M_J\} \cup \{\lambda_K^{\mathbb{N}}([m']_{e'}) \mid [m']_{e'} \in M_K\}) + 1$ ,  $\lambda_G^{+\mu} \stackrel{\text{df.}}{=} \langle \lambda_G^{\text{OP}}, \lambda_G^{\text{QA}}, n \mapsto \lambda_G^{\mathbb{N}}(n) + \mu \rangle$  for any game  $G$
- $\star \vdash_{J \ddagger K} [(m, i)]_{e_i} \stackrel{\text{df.}}{\Leftrightarrow} i = 1 \wedge \star \vdash_K [m]_e$
- $[(m, i)]_{e_i} \vdash_{J \ddagger K} [(m', j)]_{e'_j} \stackrel{\text{df.}}{\Leftrightarrow} (i = 0 = j \wedge [m]_e \vdash_J [m']_{e'}) \vee (i = 1 = j \wedge [m]_e \vdash_K [m']_{e'}) \vee (i = 1 \wedge \star \vdash_B [\pi_1(m)]_e \wedge j = 0 \wedge \star \vdash_B [\pi_1(m')]_{e'})$
- $P_{J \ddagger K} \stackrel{\text{df.}}{=} \{s \in \mathcal{J}_{J \ddagger K} \mid s \upharpoonright 0 \in P_J, s \upharpoonright 1 \in P_K, s \upharpoonright B_1, B_2 \in \text{pr}_B\}$ , where  $B_1, B_2$  are the two copies of  $B$ ,  $s \upharpoonright B_1, B_2$  is the subsequence of  $s$  consisting of moves in  $B_1, B_2$ , i.e., external moves  $[(m, i), j]_{e_{ij}}$  such that  $(i = 1 \wedge j = 0) \vee (i = 0 \wedge j = 1)$ , with the justifiers in  $s$  but changed into  $[(m, j)]_{e_j}$ ,  $\text{pr}_B \stackrel{\text{df.}}{=} \{t \in P_{B \multimap B} \mid \forall u \preceq t. \text{even}(u) \Rightarrow u \upharpoonright 0 = u \upharpoonright 1\}$

$$\blacktriangleright s \simeq_{J \ddagger K} t \stackrel{\text{df.}}{\Leftrightarrow} (\pi_2 \circ \pi_1)^*(s) = (\pi_2 \circ \pi_1)^*(t) \wedge s \upharpoonright 0 \simeq_J t \upharpoonright 0 \wedge s \upharpoonright 1 \simeq_K t \upharpoonright 1.$$

These constructions clearly preserve the axioms l1, l2, l3 (linear implication  $\multimap$  preserves l2 as games are well-opened), and so combined with the results in [YA16] we have:

**► Lemma 2.2.25** (Well-defined constructions on games). *The constructions  $\otimes$ ,  $\multimap$ ,  $\&$ ,  $!$ ,  $\ddagger$  on games are well-defined except that  $\otimes$ ,  $!$  do not preserve (finite) well-openness.*

### 2.3 Dynamic strategies

Next, let us recall the notion of *dynamic strategies* [YA16]. However, there is nothing special in the definition; a strategy  $\sigma : G$  is *dynamic* if so is  $G$ , or more precisely:

**► Definition 2.3.1** (Dynamic strategies [AM99, McC98]). A **(dynamic) strategy**  $\sigma$  on a (dynamic) game  $G$ , written  $\sigma : G$ , is a subset  $\sigma \subseteq P_G^{\text{even}}$  that satisfies:

- (S1) It is non-empty and *even-prefix-closed*:  $smn \in \sigma \Rightarrow s \in \sigma$
- (S2) It is *deterministic*:  $smn, smn' \in \sigma \Rightarrow n = n' \wedge \mathcal{I}_{smn}(n) = \mathcal{I}_{smn'}(n')$ .

**► Definition 2.3.2** (Consistent strategies). A strategy  $\sigma : G$  is **consistent** if  $\sigma \simeq_G \sigma$ , where for all  $\phi, \psi : G$ ,  $\phi \simeq_G \psi \stackrel{\text{df.}}{\Leftrightarrow} \forall s \in \phi, t \in \psi, sm, tn \in P_G. sm \simeq_G tn \Rightarrow (smm' \in \phi \Rightarrow \exists tnn' \in \psi. smm' \simeq_G tnn') \wedge (tnn'' \in \psi \Rightarrow \exists smm'' \in \phi. tnn'' \simeq_G smm'')$ .

This condition is the same as the one in [AJM00, McC98] though the word “consistency” is not used there. It ensures that strategies behave “consistently” up to permutations of tags in exponential  $!$ ; in fact, identification of positions is defined solely for consistency of strategies. For instance, a consistent strategy  $\sigma : !2$  satisfies  $[q]_{\langle f \rangle \#} [b]_{\langle f \rangle \#}, [q]_{\langle f' \rangle \#} [b']_{\langle f' \rangle \#} \in \sigma \Rightarrow b = b'$ .

As in the case of games, we define the *hiding operation on strategies*:

**► Definition 2.3.3** (Hiding operation on strategies [YA16]). For any game  $G$ ,  $s \in P_G$  and  $d \in \mathbb{N} \cup \{\omega\}$ , let  $s \ddagger \mathcal{H}_G^d \stackrel{\text{df.}}{=} \begin{cases} \mathcal{H}_G^d(s) & \text{if } s \text{ is } d\text{-complete} \\ t & \text{otherwise, where } \mathcal{H}_G^d(s) = tm. \end{cases}$  We define the ***d*-hiding operation**  $\mathcal{H}^d$  on strategies by  $\mathcal{H}^d : (\sigma : G) \mapsto \{s \ddagger \mathcal{H}_G^d \mid s \in \sigma\}$ . A strategy  $\sigma : G$  is **static** if  $\mathcal{H}^\omega(\sigma) = \sigma$ .

**► Theorem 2.3.4** (Hiding theorem [YA16]). *If  $\sigma : G$ , then  $\mathcal{H}^d(\sigma) : \mathcal{H}^d(G)$  for all  $d \in \mathbb{N} \cup \{\omega\}$ .*

Next, let us review the standard constructions on strategies [AM99, McC98], for which we need to adopt our particular implementation of tags.

**► Definition 2.3.5** (Copy-cat strategies [AJ94, AJM00, HO00, McC98]). The **copy-cat strategy**  $cp_A : A \multimap A$  on a game  $A$  is defined by  $cp_A \stackrel{\text{df.}}{=} \{s \in P_{A \multimap A}^{\text{even}} \mid \forall t \preceq s. \text{even}(t) \Rightarrow t \upharpoonright 0 = t \upharpoonright 1\}$ .

**► Definition 2.3.6** (Tensor [AJ94, McC98]). Given  $\sigma : A \multimap C, \tau : B \multimap D$ , their **tensor (product)**  $\sigma \otimes \tau : A \otimes B \multimap C \otimes D$  is defined by  $\sigma \otimes \tau \stackrel{\text{df.}}{=} \{s \in L_{A \otimes B \multimap C \otimes D} \mid s \upharpoonright [0] \in \sigma, s \upharpoonright [1] \in \tau\}$ , where  $s \upharpoonright [0]$  (resp.  $s \upharpoonright [1]$ ) is the subsequence of  $s$  with the justifiers in  $s$  that consists of moves  $[((m, 0), i)]_{e_{0i}}$  (resp.  $[((m', 1), j)]_{e_{1j}}$ ) changed into  $[((m, i)]_{e_i}$  (resp.  $[((m', j)]_{e_j}$ ).

**► Definition 2.3.7** (Pairing [AJM00, McC98]). Given  $\sigma : C \multimap A, \tau : C \multimap B$ , their **pairing**  $\langle \sigma, \tau \rangle : C \multimap A \& B$  is defined by  $\langle \sigma, \tau \rangle \stackrel{\text{df.}}{=} \{s \in L_{C \multimap A \& B} \mid s \upharpoonright ([0] \multimap [01]) \in \sigma, s \upharpoonright ([0] \multimap [11]) = \epsilon\} \cup \{s \in L_{C \multimap A \& B} \mid s \upharpoonright ([0] \multimap [11]) \in \tau, s \upharpoonright ([0] \multimap [01]) = \epsilon\}$ , where  $s \upharpoonright ([0] \multimap [01])$  (resp.  $s \upharpoonright ([0] \multimap [11])$ ) is the subsequence of  $s$  with the justifiers in  $s$  that consists of moves  $[((c, 0)]_{e_0}$ ,  $[((a, 0), 1)]_{e'_{01}}$  (resp.  $[((b, 1), 1)]_{e'_{11}}$ ) hereditarily justified by  $[((a', 0), 1)]$  for some  $[a'] \in M_A^{\text{init}}$  (resp.  $[((b', 1), 1)]$  for some  $[b'] \in M_B^{\text{init}}$ ) with the latter changed into  $[((a, 1)]_{e'_1}$  (resp.  $[((b, 1)]_{e'_1}$ ).

► **Definition 2.3.8** (Generalized pairing [YA16]). Given  $\sigma : L, \tau : R$  such that  $\mathcal{H}^\omega(L) \trianglelefteq C \multimap A, \mathcal{H}^\omega(R) \trianglelefteq C \multimap B$  for some static games  $A, B, C$ , their **(generalized) pairing**  $\langle \sigma, \tau \rangle : L \& R$  is defined by  $\langle \sigma, \tau \rangle \stackrel{\text{df.}}{=} \{s \in L \& R \mid s \upharpoonright L \in \sigma, s \upharpoonright R \in \tau\} \cup \{s \in L \& R \mid s \upharpoonright L = \epsilon, s \upharpoonright R \in \tau\}$ .

► **Definition 2.3.9** (Promotion [AJM00, McC98]). Given  $\sigma : !A \multimap B$ , its **promotion**  $\sigma^\dagger : !A \multimap !B$  is defined by  $\sigma^\dagger \stackrel{\text{df.}}{=} \{s \in P_{!A \multimap !B} \mid \forall e \in \mathcal{T}. s \upharpoonright e \in \sigma\}$ , where  $s \upharpoonright e$  is the subsequence of  $s$  with the justifiers in  $s$  that consists of moves  $[(b, 1)]_{((e) \# e')_1}, [(a, 0)]_{((\langle e \rangle \# \langle f \rangle) \# f')_0}$ , where  $[b]_{e'} \in M_B, [a]_{f'} \in M_A$ , changed into  $[(b, 1)]_{e'_1}, [(a, 0)]_{(\langle f \rangle \# f')_0}$ , respectively.

► **Definition 2.3.10** (Dereliction [AJM00, McC98]). Let  $A$  be a well-opened game. The **dereliction**  $der_A : A \Rightarrow A$  on  $A$  is defined by  $der_A \stackrel{\text{df.}}{=} \{s \in P_{A \Rightarrow A}^{\text{even}} \mid \forall t \preceq s. \text{even}(t) \Rightarrow t \upharpoonright [0]_{\langle \rangle \#_-} = t \upharpoonright [1]_{-}\}$ , where  $t \upharpoonright [0]_{\langle \rangle \#_-}$  (resp.  $t \upharpoonright [1]_{-}$ ) is the subsequence of  $t$  with the same justifiers in  $t$  that consists of moves  $[(a, 0)]_{(\langle \rangle \# e)_0}$  (resp.  $[(a', 1)]_{e'_1}$ ) changed into  $[a]_e$  (resp.  $[a']_{e'}$ ).

► **Definition 2.3.11** (Concatenation and composition [YA16]). Let  $\sigma : J, \tau : K$ , and assume that  $\mathcal{H}^\omega(J) \trianglelefteq A \multimap B, \mathcal{H}^\omega(K) \trianglelefteq B \multimap C$  for some static games  $A, B, C$ . Their **concatenation**  $\sigma \ddagger \tau : J \ddagger K$  is defined by  $\sigma \ddagger \tau \stackrel{\text{df.}}{=} \{s \in \mathcal{J}_{J \ddagger K} \mid s \upharpoonright 0 \in \sigma, s \upharpoonright 1 \in \tau, s \upharpoonright B_1, B_2 \in \text{pr}_B\}$  and their **composition**  $\sigma; \tau : \mathcal{H}^\omega(J \ddagger K)$  is defined by  $\sigma; \tau \stackrel{\text{df.}}{=} \mathcal{H}^\omega(\sigma \ddagger \tau)$ .

If  $J = A \multimap B, K = B \multimap C$ , then our composition  $\sigma; \tau : \mathcal{H}^\omega(A \multimap B \ddagger B \multimap C) \trianglelefteq A \multimap C$  (this relation holds only up to tags) coincides with the standard one in the literature [HO00, AM99, McC98]; see [YA16] for the details.

### 3 Effective strategies

We have presented our variant of games and strategies. In this main section, we introduce an *intrinsic* notion of “effective computability” of strategies that subsumes computation of the programming language PCF [Plo77, Mit96], and so it is *Turing complete* in particular.

► **Notation.** We often write  $A \Rightarrow B$  or  $A \rightarrow B$  for the linear implication  $!A \multimap B$  for any games  $A, B$ . The operations  $\multimap, \Rightarrow, \rightarrow$  are all right associative.

#### 3.1 Effective strategies

As *history-free* strategies are expressive enough to model the language PCF [AJM00], it suffices for our strategies to refer to at most *three last moves* in the P-view and the (“semi”) *opening moves* of the position, which is clearly “effective”. Thus, it remains to formulate the notion of “effective computability” of the next move from such a bounded number of previous moves.

Since the set  $\{m \mid [m]_e \in M_G\}$  is finite for any game  $G$ , *finitary* (innocent [HO00, AM99, McC98]) strategies in the sense that their *view functions* [HO00, McC98] are finite seem sufficient at first glance. However, to model the *fixed-point combinators* in PCF, strategies need to be able to initiate new threads *unboundedly many times* [HO00, AJM00]; also, they have to model *promotion*  $(\_)^\dagger$  for which infinitely many manipulations of tags are necessary. Thus, finitary strategies are not strong enough.

Then how can we define a stronger notion of “effective computability” of the next move from previous moves *solely in terms of games and strategies*? Our solution is as follows. A strategy  $\sigma : G$  is *effective* if it is “describable” by a finitary strategy on the *instruction game*:

► **Definition 3.1.1** (Instruction games). Given a game  $G$ , its **instruction game**  $\mathcal{G}(M_G)$  is the product  $\mathcal{G}(\pi_1(M_G)) \& \mathcal{G}(\mathcal{T})$ , where the component game  $\mathcal{G}(\pi_1(M_G))$  is defined by:

- $M_{\mathcal{G}(\pi_1(M_G))} \stackrel{\text{df.}}{=} \{q_G, \square\} \cup \pi_1(M_G)$ , where  $q_G, \square$  are arbitrarily chosen with  $q_G \notin \pi_1(M_G)$ ,  $\square \notin \pi_1(M_G)$ , and each move has the empty tag  $\epsilon$
- $\lambda_{\mathcal{G}(\pi_1(M_G))} : q_G \mapsto (O, Q, 0), (m \in \pi_1(M_G)) \mapsto (P, A, 0), \square \mapsto (P, A, 0)$
- $\vdash_{\mathcal{G}(\pi_1(M_G))} \stackrel{\text{df.}}{=} \{(\star, q_G), (q_G, \square)\} \cup \{(q_G, m) \mid m \in \pi_1(M_G)\}$
- $P_{\mathcal{G}(\pi_1(M_G))} \stackrel{\text{df.}}{=} \text{pref}(\{q_G.m \mid m \in \pi_1(M_G)\} \cup \{q_G.\square\})$ , where  $q_G$  justifies  $m$  and  $\square$
- $s \simeq_{\mathcal{G}(\pi_1(M_G))} t \stackrel{\text{df.}}{\iff} s = t$ .

The positions  $q_G.m, q_G.\square$  are to represent  $m \in \pi_1(M_G)$ , “no element”, respectively.

- **Notation.** Given a sequence  $s = x_k x_{k-1} \dots x_1 \in M_G^*$  of moves in an arena  $G$  and a number  $l \in \mathbb{N}$ , we define  $s \downarrow l \stackrel{\text{df.}}{=} \begin{cases} s & \text{if } l \geq k \\ x_l x_{l-1} \dots x_1 & \text{otherwise.} \end{cases}$  A function  $f : \pi_1(M_G) \rightarrow \{\top, \perp\}$  induces

another  $f^* : M_G^* \rightarrow \pi_1(M_G)^*$  by  $f([m_k]_{e_k} [m_{k-1}]_{e_{k-1}} \dots [m_1]_{e_1}) \stackrel{\text{df.}}{=} m_{i_l} m_{i_{l-1}} \dots m_{i_1}, l \leq k$ , where  $m_{i_l} m_{i_{l-1}} \dots m_{i_1}$  is a subsequence of  $m_k m_{k-1} \dots m_1$  that consists of  $m_{i_j}$  such that  $f(m_{i_j}) = \top$ .

- **Notation.** Let  $G$  be a game, and  $[m]_e \in M_G, e = e_1 e_2 \dots e_k \in \mathcal{T}$ . We write  $[m]_e$  for the strategy  $\langle \underline{m}, \underline{e} \rangle : \mathcal{G}(M_G)$ , where  $\underline{m} : \mathcal{G}(\pi_1(M_G))$ ,  $\underline{e} : \mathcal{G}(\mathcal{T})$  are defined by  $\underline{m} \stackrel{\text{df.}}{=} \text{pref}(\{q_G.m\})^{\text{even}}$ ,  $\underline{e} \stackrel{\text{df.}}{=} \text{pref}(\{q_{\mathcal{T}} e_1 q_{\mathcal{T}} e_2 \dots q_{\mathcal{T}} e_k q_{\mathcal{T}} \checkmark\})^{\text{even}}$ , respectively. Similarly, we define  $\square \stackrel{\text{df.}}{=} \text{pref}(\{q_G.\square\})^{\text{even}} : \mathcal{G}(\pi_1(M_G))$ , and  $\underline{\square} \stackrel{\text{df.}}{=} \langle \underline{\square}, \underline{e} \rangle : \mathcal{G}(M_G)$ . For any  $s = [m_l]_{e_l} [m_{l-1}]_{e_{l-1}} \dots [m_1]_{e_1} \in M_G^*, n \geq l$ , we define  $\underline{s}_n \stackrel{\text{df.}}{=} \underbrace{\langle \underline{\square}, \dots, \underline{\square} \rangle}_{n-l}, \underbrace{[m_l]_{e_l}, [m_{l-1}]_{e_{l-1}}, \dots, [m_1]_{e_1}}_n : \mathcal{G}(M_G)^n \stackrel{\text{df.}}{=} \underbrace{\mathcal{G}(M_G) \& \dots \& \mathcal{G}(M_G)}_n$ ,

where the pairing and product are left associative. Given a strategy  $\sigma : \mathcal{G}(M_G)$ , we define  $\mathcal{M}(\sigma) \in M_G$  to be the unique move such that  $\mathcal{M}(\sigma) = \sigma$  if it exists, and undefined otherwise.

We will be particularly concerned with games  $P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}$  shortly. The symbols  $\langle, \rangle$  in any position  $s \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}$  form *unique* pairs similarly to “QA-pairs” for *bracketing condition* [HO00, AM99]. Specifically, each  $\rangle$  is paired with the most recent “still unpaired”  $\langle$  in the same component game  $\mathcal{G}(\mathcal{T})$ ; one is called the *mate* of the other. Moreover, we define:

- **Definition 3.1.2** (M-views). Let  $G$  be a game. The *matching view (m-view)*  $\llbracket s \rrbracket_G$  (we often abbreviate it as  $\llbracket s \rrbracket$ ) of a position  $s \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}$  is defined by:  $\llbracket \epsilon \rrbracket_G \stackrel{\text{df.}}{=} \epsilon$ ,  $\llbracket s.\langle t. \rangle \rrbracket_G \stackrel{\text{df.}}{=} \llbracket s \rrbracket_G.\langle \cdot \rangle$ , where  $\langle$  is the mate of  $\rangle$ , and  $\llbracket sm \rrbracket_G \stackrel{\text{df.}}{=} \llbracket s \rrbracket_G.m$ , where  $m \neq \rangle$ .

We are now ready to make the notion of “describable by a finitary strategy” precise.

- **Definition 3.1.3** (Algorithms). An *algorithm*  $\mathcal{A}$  on a game  $G$ , written  $\mathcal{A} :: G$ , is a collection  $\mathcal{A} = (\mathcal{A}_m)_{m \in \mathcal{S}_{\mathcal{A}}}$  of finite partial functions  $\mathcal{A}_m : \partial_m(P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}^{\text{odd}}) \rightarrow M_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}$ , where  $\mathcal{S}_{\mathcal{A}} \subseteq \pi_1(M_G)^* \setminus \{\epsilon\}$  is a finite set of *states*,  $\partial_m(tx) \stackrel{\text{df.}}{=} (\mathcal{O}(\lceil tx \rceil), \lceil tx \rceil \downarrow |\mathcal{A}_m|, \lceil tx \rceil \downarrow \|\mathcal{A}_m\|)$  for all  $tx \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}^{\text{odd}}$ ,  $|\mathcal{A}_m|, \|\mathcal{A}_m\| \in \mathbb{N}$  are the *scopes* of  $\mathcal{A}_m$ , and  $\mathcal{A}_m$  also specifies the justifier of each output in the input<sup>2</sup>, equipped with the *query (function)*  $\mathcal{Q}_{\mathcal{A}} : \pi_1(M_G) \rightarrow \{\top, \perp\}$  such that  $[m] \in M_G^{\text{init}} \Rightarrow \mathcal{Q}_{\mathcal{A}}(m) = \top$  and  $\mathcal{Q}_{\mathcal{A}}(m) = \top \Rightarrow \exists e \in \mathcal{T}. [m]_e \in M_G$  is initial or internal.

- **Remark.** Note that  $\mathcal{Q}_{\mathcal{A}}^*(s) \neq \epsilon$  for any  $s \in \mathcal{S}_{\mathcal{A}} \setminus \{\epsilon\}$  since  $[m] \in M_G^{\text{init}} \Rightarrow \mathcal{Q}_{\mathcal{A}}(m) = \top$ .

<sup>2</sup>But we usually treat this structure *implicit* as in [AM99, McC98].

► **Convention.** Since an algorithm  $\mathcal{A} :: G$  takes into account the opening move  $\mathcal{O}(\lceil tx \rceil)$  for each  $tx \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}^{\text{odd}}$  *only occasionally*, we usually take  $\partial_m(tx) \stackrel{\text{df.}}{=} (\lceil tx \rceil \downarrow |\mathcal{A}_m|, \llbracket tx \rrbracket \downarrow \|\mathcal{A}_m\|)$  for all  $tx \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}^{\text{odd}}$  for each  $m \in \mathcal{S}_{\mathcal{A}}$ , keeping in mind that  $\mathcal{A}_m$  sees opening moves.

► **Definition 3.1.4** (Instruction strategies). Given a game  $G$  and an algorithm  $\mathcal{A} :: G$ , we define the *instruction strategy*  $\mathcal{A}_m^* : \mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)$  by:

$$\mathcal{A}_m^* \stackrel{\text{df.}}{=} \{\epsilon\} \cup \{txy \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)} \mid t \in \mathcal{A}_m^*, tx \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}, y = \mathcal{A}_m(\partial_m(tx))\}.$$

► **Convention.** Only in a limited situation, an algorithm  $\mathcal{A}$  takes into account a bounded size  $\|\mathcal{A}_m\| \in \mathbb{N}$  of information from  $m$ -views. Thus, in most cases, each  $\mathcal{A}_m$  is a partial function  $\mathcal{A}_m : \{\lceil tx \rceil \downarrow |\mathcal{A}_m| \mid tx \in P_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}^{\text{odd}}\} \rightarrow M_{\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)}$ ; we in fact treat it as such unless necessary. Accordingly,  $\mathcal{A}_m^*$  is usually a strategy  $\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)$  whose *view function* representation [HO00, McC98]  $\mathcal{A}_m$  is finite, where the *scope*  $|\mathcal{A}_m|$  is to keep inputs *finite*.<sup>3</sup>

► **Notation.**  $\simeq$  denotes the *Kleene equality* [TVD14], i.e.,  $x \simeq y \stackrel{\text{df.}}{\iff} (x \downarrow \wedge y \downarrow \wedge x = y) \vee (x \uparrow \wedge y \uparrow)$ .

► **Definition 3.1.5** (Realizability). The strategy  $\text{st}(\mathcal{A})$  *realized* by  $\mathcal{A} :: G$  is defined by  $\text{st}(\mathcal{A}) \stackrel{\text{df.}}{=} \{\epsilon\} \cup \{sa.\mathcal{A}^*(\lceil sa \rceil \downarrow 3) \mid s \in \text{st}(\mathcal{A}), sa \in P_G, sa.\mathcal{A}^*(\lceil sa \rceil \downarrow 3) \in P_G\}$ , where  $\mathcal{A}^*(\lceil sa \rceil \downarrow 3) \stackrel{\text{df.}}{\simeq} \mathcal{M}(\mathcal{A}_{\mathcal{A}^*(\lceil sa \rceil \downarrow 3)}^* \circ \lceil sa \rceil \downarrow 3_3^\dagger)$ , and  $\mathcal{A}^*(\lceil sa \rceil \downarrow 3) \in P_G$  presupposes  $\mathcal{Q}_{\mathcal{A}}^*(\lceil sa \rceil) \in \mathcal{S}_{\mathcal{A}} \wedge \mathcal{A}^*(\lceil sa \rceil \downarrow 3) \downarrow$ .

► **Remark.** Strictly speaking, each  $\mathcal{A}_m^* : \mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)$  has to specify justifiers (in  $P_G$ ) of outputs. This is easily achieved by changing it into  $\mathcal{A}_m^* : \mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G) \& \mathbf{2}$  as the choice is *ternary* (the last or third last move in the P-view, or the opening move). However, since justifiers in this paper are obvious ones, we adopt the abbreviated form of  $\mathcal{A}_m^*$  as above.

Clearly,  $\mathcal{A} :: G \Rightarrow \text{st}(\mathcal{A}) : G$  holds. We are now ready to define the central notion of the paper, namely “*effective computability*” of strategies, in an *intrinsic* manner:

► **Definition 3.1.6** (Effective strategies). A strategy  $\sigma : G$  is *effective* if there exists an algorithm  $\mathcal{A} :: G$  that realizes  $\sigma$ , i.e.,  $\text{st}(\mathcal{A}) = \sigma$ .

Given an algorithm  $\mathcal{A} :: G$  that realizes an effective strategy  $\sigma : G$ , we may “*effectively execute*”  $\mathcal{A}$  to compute  $\sigma$  roughly as follows:

1. Given  $sa \in P_G^{\text{odd}}$ , we calculate  $m \stackrel{\text{df.}}{=} \mathcal{Q}_{\mathcal{A}}^*(\lceil sa \rceil)$  and  $\lceil sa \rceil \downarrow 3$ . If  $m \notin \mathcal{S}_{\mathcal{A}}$ , then we stop.
2. Otherwise, we compose  $\lceil sa \rceil \downarrow 3_3^\dagger$  with  $\mathcal{A}_m^*$ , and “*execute*”  $\mathcal{A}_m^* \circ \lceil sa \rceil \downarrow 3_3^\dagger$ .
3. Finally, we “*read off*” the next move  $\mathcal{M}(\mathcal{A}_m^* \circ \lceil sa \rceil \downarrow 3_3^\dagger)$  (and its justifier).

This procedure is similar to the execution of *Turing machines* [Tur36], and intuitively “*effective*”, which is our conceptual justification of our notion of “*effective computability*”.

► **Notation.** To describe a finite partial function  $f$ , we list every input/output pair  $(x, y) \in f$  as  $f : x_0 \mapsto y_0 \mid x_1 \mapsto y_1 \mid \dots$ . Given a game  $G$ , we abuse notation and write  $m_{i_1 i_2 \dots i_k}$  as the *symbol* to denote each  $(\dots((m, i_1), i_2), \dots, i_k) \in \pi_1(M_G)$ . We often indicate the form of tags of moves  $[m_{i_1 i_2 \dots i_k}]_e$  in a game  $G$  by  $[G_{i_1 i_2 \dots i_k}]_e$ , where we call  $m_{i_1 i_2 \dots i_k}$  the *inner element*,  $i_1 i_2 \dots i_k$  and  $e$  the (*inner* and *outer*) *tags* of  $[m_{i_1 i_2 \dots i_k}]_e$ , respectively. However, we usually write tags in  $\mathcal{G}(M_G)^3 \Rightarrow \mathcal{G}(M_G)$  *informally* for brevity (which is “*harmless*” as instruction strategies are *finitary*), e.g.,  $\mathcal{G}(M_G)_0 \Rightarrow \mathcal{G}(M_G)_1, \mathcal{G}(M_G)_0 \& \mathcal{G}(M_G)_1 \& \mathcal{G}(M_G)_2 \Rightarrow \mathcal{G}(M_G)_3, [q_G]_0, [q_T]_1$ .

<sup>3</sup>The point here is that instruction strategies are clearly “*computable*”, but they achieve an *unbounded* collection of manipulations of tags.

► **Convention.** Since we shall focus on *consistent* strategies of the form  $A \Rightarrow B$  in this paper, it is reasonable to require each algorithm  $\mathcal{A}$  not to refer to any outer tags when it computes the next internal element. Also, since a strategy of our interest modifies (i.e., not just “copy-cats”) the outer tag of the last move only if it initiates a new thread in the domain game, we assume that the only outer tag  $\mathcal{A}$  investigates when it computes the next outer tag is just the one of the last move in the P-view, and it reads it off at most once. We call algorithms that satisfy these two conditions *standard*; from now on, the word *algorithms* refers to *standard* ones by default. Algorithms in this paper are all standard, and standard algorithms are closed under all constructions on algorithms we shall introduce later. This convention will save work in the proof to show the closure property of effective strategies under promotion (see Theorem 3.1.9).

► **Example 3.1.7.** The *zero strategy*  $zero \stackrel{\text{df.}}{=} \text{pref}(\{[q_1][b_1]\})^{\text{even}} : [I_0]_{\langle e \rangle \#} \Rightarrow [N_1]$  is effective since we may give an algorithm  $\mathcal{A}(zero)$  by  $\mathcal{Q}_{\mathcal{A}(zero)}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_1 \\ \perp & \text{otherwise} \end{cases}, \mathcal{S}_{\mathcal{A}(zero)} \stackrel{\text{df.}}{=} \{q_1\}, |\mathcal{A}(zero)_{q_1}| \stackrel{\text{df.}}{=} 1$  and  $\mathcal{A}(zero)_{q_1} : [q_{I \Rightarrow N}]_1 \mapsto [b_1]_1 \mid [q_{\mathcal{T}}]_1 \mapsto [\checkmark]_1$ . Then the instruction strategy  $\mathcal{A}(zero)_{q_1}^*$  is as depicted in the following diagram:

$$\frac{\mathcal{G}(M_{I \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{I \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{I \Rightarrow N})_0}{\mathcal{G}(M_{I \Rightarrow N})_1} \xRightarrow{\mathcal{A}(zero)_{q_1}^*} \frac{\mathcal{G}(M_{I \Rightarrow N})_1}{\begin{array}{c} [q_{I \Rightarrow N}]_1 \ ([q_{\mathcal{T}}]_1) \\ [b_1]_1 \ ([\checkmark]_1) \end{array}}$$

Clearly,  $\text{st}(\mathcal{A}(zero)) = zero$ .

Next, let us consider the *successor strategy*

$$succ \stackrel{\text{df.}}{=} \text{pref}(\{[q_1][q_0]_{\langle \rangle \#} ([\bullet_0]_{\langle \rangle \#} [\bullet_1][q_1][q_0]_{\langle \rangle \#})^n [b_0]_{\langle \rangle \#} [\bullet_1][q_1][b_1] \mid n \in \mathbb{N}\})^{\text{even}} : [N_0]_{\langle e_0 \rangle \#} \Rightarrow [N_1].$$

We give an algorithm  $\mathcal{A}(succ)$  for  $succ$  by defining  $\mathcal{Q}_{\mathcal{A}(succ)}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_1 \\ \perp & \text{otherwise} \end{cases}, \mathcal{S}_{\mathcal{A}(succ)} \stackrel{\text{df.}}{=} \{q_1\}, |\mathcal{A}(succ)_{q_1}| \stackrel{\text{df.}}{=} 5$ , and

$$\begin{aligned} \mathcal{A}(succ)_{q_1} : [q_{N \Rightarrow N}]_3 \mapsto [q_{N \Rightarrow N}]_2 \mid [q_{N \Rightarrow N}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 \mapsto [q_{N \Rightarrow N}]_0 \mid \\ [q_{N \Rightarrow N}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 [q_{N \Rightarrow N}]_0 [x]_0 \mapsto [q_0]_3 \mid \\ [q_{N \Rightarrow N}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 [q_{N \Rightarrow N}]_0 [b_0]_0 \mapsto [b_1]_3 \mid [q_{\mathcal{T}}]_3 \mapsto [q_{N \Rightarrow N}]_2 \mid \\ [q_{\mathcal{T}}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 \mapsto [q_{N \Rightarrow N}]_0 \mid [q_{\mathcal{T}}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 [q_{N \Rightarrow N}]_0 [x]_0 \mapsto [\langle \rangle]_3 \mid \\ [q_1]_2 [q_{N \Rightarrow N}]_0 [x]_0 [\langle \rangle]_3 [q_{\mathcal{T}}]_3 \mapsto [\rangle]_3 \mid [x]_0 [\langle \rangle]_3 [q_{\mathcal{T}}]_3 [\rangle]_3 [q_{\mathcal{T}}]_3 \mapsto [\#]_3 \mid \\ [q_{\mathcal{T}}]_3 [\rangle]_3 [q_{\mathcal{T}}]_3 [\#]_3 [q_{\mathcal{T}}]_3 \mapsto [\checkmark]_3 \mid [q_{\mathcal{T}}]_3 [q_{N \Rightarrow N}]_2 [q_1]_2 [q_{N \Rightarrow N}]_0 [b_0]_0 \mapsto [\checkmark]_3 \mid \\ [q_{N \Rightarrow N}]_3 [q_{N \Rightarrow N}]_2 [y]_2 \mapsto [\bullet_1]_3 \mid [q_{\mathcal{T}}]_3 [q_{N \Rightarrow N}]_2 [y]_2 \mapsto [\checkmark]_3 \end{aligned}$$

where  $x \in \{\square, \bullet_0\}$   $y \in \{\bullet_0, b_0\}$ . Consequently,  $\mathcal{A}(succ)_{q_1}^*$  is as depicted in the following:

$$\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2}{\mathcal{G}(M_{N \Rightarrow N})_3} \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \frac{\mathcal{G}(M_{N \Rightarrow N})_3}{\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [q_1]_2 \\ [q_{N \Rightarrow N}]_0 \\ [\square]_0 \ ([\bullet_0]_0) \end{array}} \quad [q_0]_3$$



$$\begin{array}{c}
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{N \Rightarrow N}]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_0 \\ [b_0]_0 \end{array} \quad \begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [q_1]_2 \end{array} \quad [b_1]_3 \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q\tau]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_0 \\ [\square]_0 \ (\bullet_0)_0 \end{array} \quad \begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [q_1]_2 \end{array} \quad \begin{array}{c} [\langle ]_3 \\ [q\tau]_3 \\ [ ]_3 \\ [q\tau]_3 \\ [\sharp]_3 \\ [q\tau]_3 \\ [\checkmark]_3 \end{array} \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q\tau]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_0 \\ [b_0]_0 \end{array} \quad \begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [q_1]_2 \end{array} \quad [\checkmark]_3 \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{N \Rightarrow N}]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [\bullet_0]_2 \ (\llbracket b_0 \rrbracket_2) \end{array} \quad [\bullet_1]_3 \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(succ)_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q\tau]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [\bullet_0]_2 \ (\llbracket b_0 \rrbracket_2) \end{array} \quad [\checkmark]_3
\end{array}$$

We clearly have  $\text{st}(\mathcal{A}(succ)) = succ$ , which establishes the effectivity of  $succ$ .

► **Example 3.1.8.** Consider the *fixed-point strategy*  $fix_A : ([A_{00}]_{\langle e' \rangle \sharp (e) \sharp f} \Rightarrow [A_{10}]_{\langle e' \rangle \sharp f}) \Rightarrow [A_1]_f$  for each game  $A$  that interprets the *fixed-point combinator*  $\text{fix}_A$  in PCF [AJM00, HO00, McC98]. Roughly,  $fix_A$  computes as follows (for its detailed description, see [Hyl97, HO00]):

1. After the first O-move  $[a_1]$ ,  $fix_A$  copies it and makes the second move  $[a_{10}]_{\langle \rangle \sharp}$ , where note that the first move must have the empty outer tag  $\epsilon$  as  $A$  is finitely well-opened.

2. If Opponent initiates a new thread  $[a'_{00}]_{\langle e' \rangle \# \langle e \rangle \# f}$  in the inner implication, then  $fix_A$  copies it and launches a new thread in the outer implication by  $[a'_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$ .
3. If Opponent makes a move  $[a''_{00}]_{\langle e' \rangle \# \langle e \rangle \# f}$  (resp.  $[a''_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$ ,  $[a''_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$ ,  $[a''_{11}]_f$ ) in an existing thread, then  $fix_A$  copies it and makes the next move  $[a'_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$  (resp.  $[a'_{11}]_f$ ,  $[a'_{00}]_{\langle e' \rangle \# \langle e \rangle \# f}$ ,  $[a'_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$ ) in the “dual thread” (to which the third last move belongs).

Clearly,  $fix_A$  is not finitary for the calculation of outer tags. It is, however, effective for any game  $A$ , which is perhaps surprising to many readers. Here, let us just informally describe an algorithm  $\mathcal{A}(fix_A)$  that realizes  $fix_A$  (see Section 3.2 for the detailed treatment):

- $\mathcal{Q}_{\mathcal{A}(fix_A)}(m) = \top$  iff  $[m]$  is initial, and  $\mathcal{S}_{\mathcal{A}(fix_A)} = \{m \in \pi_1(M_{(A \Rightarrow A) \Rightarrow A}) \mid \star \vdash_{(A \Rightarrow A) \Rightarrow A} [m]\}$ . Since  $\mathcal{A}(fix_A)_m$  does not depend on  $m$ , fix an arbitrary  $m$  such that  $[m]$  initial.
- If the rightmost component of the input strategy for  $\mathcal{A}(fix_A)_m^*$  is of the form  $([a_1]_f)^\dagger$ , then  $\mathcal{A}(fix_A)_m^*$  calculates the next move  $[a_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$  once and for all for the internal element and “digit-by-digit” for the outer tag.
- If the rightmost component is of the form  $([a_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f})^\dagger$ , then  $\mathcal{A}(fix_A)_m^*$  recognizes it by investigating the third rightmost component, and calculates the next move  $[a_1]_f$  once and for all for the internal element and “digit-by-digit” for the outer tag.
- If the rightmost component is of the form  $([a_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f})^\dagger$ , then  $\mathcal{A}(fix_A)_m^*$  calculates the next move  $[a_{00}]_{\langle e' \rangle \# \langle e \rangle \# f}$  in a similar manner to the above case but with the help of m-views for the outer tag; see Section 3.2 for the details.
- If the rightmost component is of the form  $([a_{00}]_{\langle e' \rangle \# \langle e \rangle \# f})^\dagger$ , then  $\mathcal{A}(fix_A)_m^*$  calculates the next move  $[a_{10}]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}$  in a similar manner to the first case with the help of m-views for the outer tag (n.b. the justifier may not be the last or third last move in the P-view, but in that case it is the opening move  $[m]$ ); see Section 3.2 for the details.

► **Theorem 3.1.9** (Constructions on effective strategies). *Consistent and effective strategies are closed under tensor  $\otimes$ , pairing  $\langle -, - \rangle$ , promotion  $(-)^{\dagger}$  and concatenation  $\ddagger$ .*

*Proof.* The preservation of consistency is straightforward as in [McC98]. We first show that tensor  $\otimes$  preserves effectivity of strategies. Let  $\sigma : [A_0]_e \multimap [C_1]_{e'}$ ,  $\tau : [B_0]_f \multimap [D_1]_{f'}$  be effective strategies with algorithms  $\mathcal{A}(\sigma)$ ,  $\mathcal{A}(\tau)$  realizing  $\sigma$ ,  $\tau$ , respectively. We have to construct an algorithm  $\mathcal{A}(\sigma \otimes \tau)$  such that  $\text{st}(\mathcal{A}(\sigma \otimes \tau)) = \sigma \otimes \tau : [A_{00}]_e \otimes [B_{10}]_f \multimap [C_{01}]_{e'} \otimes [D_{11}]_{f'}$ . Define the set  $\mathcal{S}_{\mathcal{A}(\sigma \otimes \tau)}$  of states and the query  $\mathcal{Q}_{\mathcal{A}(\sigma \otimes \tau)} : \pi_1(M_{A \otimes B \multimap C \otimes D}) \rightarrow \{\top, \perp\}$  by:

$$\begin{aligned} \mathcal{S}_{\mathcal{A}(\sigma \otimes \tau)} &\stackrel{\text{df.}}{=} \{m_{0i_k}^{(k)} m_{0i_{k-1}}^{(k-1)} \dots m_{0i_1}^{(1)} \mid m_{i_k}^{(k)} m_{i_{k-1}}^{(k-1)} \dots m_{i_1}^{(1)} \in \mathcal{S}_{\mathcal{A}(\sigma)}\} \\ &\quad \cup \{n_{1j_l}^{(l)} n_{1j_{l-1}}^{(l-1)} \dots n_{1j_1}^{(1)} \mid n_{j_l}^{(l)} n_{j_{l-1}}^{(l-1)} \dots n_{j_1}^{(1)} \in \mathcal{S}_{\mathcal{A}(\tau)}\} \\ \mathcal{Q}_{\mathcal{A}(\sigma \otimes \tau)} &: a_{00} \mapsto \mathcal{Q}_{\mathcal{A}(\sigma)}(a_0) \mid b_{10} \mapsto \mathcal{Q}_{\mathcal{A}(\tau)}(b_0) \mid c_{01} \mapsto \mathcal{Q}_{\mathcal{A}(\sigma)}(c_1) \mid d_{11} \mapsto \mathcal{Q}_{\mathcal{A}(\tau)}(d_1). \end{aligned}$$

Note that  $\mathcal{Q}_{\mathcal{A}(\sigma \otimes \tau)}$  clearly satisfies the required condition, i.e., it outputs  $\top$  if the input is initial, and if it outputs  $\top$ , then the input is initial or internal. Now, construct the finite partial functions  $\mathcal{A}(\sigma \otimes \tau)_{m_{0i_k}^{(k)} m_{0i_{k-1}}^{(k-1)} \dots m_{0i_1}^{(1)}}, \mathcal{A}(\sigma \otimes \tau)_{n_{1j_l}^{(l)} n_{1j_{l-1}}^{(l-1)} \dots n_{1j_1}^{(1)}}$  from  $\mathcal{A}(\sigma)_{m_{i_k}^{(k)} m_{i_{k-1}}^{(k-1)} \dots m_{i_1}^{(1)}}$ ,  $\mathcal{A}(\tau)_{n_{j_l}^{(l)} n_{j_{l-1}}^{(l-1)} \dots n_{j_1}^{(1)}}$  simply by changing symbols of the form  $m_i$  into  $m_{0i}$ ,  $m_{1i}$  (including ones for tags) respectively in their (finite) tables. Since P-views of  $\sigma$  and  $\tau$  never interact to each other in  $\sigma \otimes \tau$  (which is shown by induction on the length of positions), it is straightforward to see that  $\text{st}(\mathcal{A}(\sigma \otimes \tau)) =$

$\sigma \otimes \tau$  holds. Intuitively,  $\mathcal{A}(\sigma \otimes \tau)$  sees the new digit (0 or 1) of the current state  $s \in \mathcal{S}_{\mathcal{A}(\sigma \otimes \tau)}$  and decides  $\mathcal{A}(\sigma)$  or  $\mathcal{A}(\tau)$  to apply (n.b.  $\mathcal{Q}_{\mathcal{A}(\sigma \otimes \tau)}$  “tracks” every initial move, and so a possible state must be non-empty in the non-trivial case<sup>4</sup>, and so it indicates the component game “at work”). Note that tags are also distinguished in this manner as each component game uses a distinguished copy of the symbols  $|, \#$ , and we distinguish them by 0, 1 digits.

Next, consider the pairing  $\langle \phi, \psi \rangle : L \& R$  of effective strategies  $\phi : L, \psi : R$  such that  $\mathcal{H}^\omega(L) = C \multimap A, \mathcal{H}^\omega(R) = C \multimap B$  for some static games  $A, B, C$ . Let  $\mathcal{A}(\phi), \mathcal{A}(\psi)$  be algorithms realizing  $\phi, \psi$ , respectively. Note that  $L \& R$  is the *generalized pairing* defined in [YA16]; roughly, it is the usual pairing but moves in  $C$  are not “duplicated”. Since the query functions  $\mathcal{Q}_{\mathcal{A}(\phi)}, \mathcal{Q}_{\mathcal{A}(\psi)}$  “track” only initial or internal moves, they in particular “ignore” moves in  $C$ . Thus, we may safely apply the same construction of algorithms as that for  $\otimes$  except that the additional 0, 1 digits lie on the righthand side, and inner tags of moves in  $C$  are not changed.

Now, consider the concatenation  $\iota \ddagger \kappa : J \ddagger K$  of effective strategies  $\iota : J, \kappa : K$  such that  $\mathcal{H}^\omega(J) = A \multimap B, \mathcal{H}^\omega(K) = B \multimap C$  for some static games  $A, B, C$ . Let  $\mathcal{A}(\iota), \mathcal{A}(\kappa)$  be algorithms such that  $\text{st}(\mathcal{A}(\iota)) = \iota, \text{st}(\mathcal{A}(\kappa)) = \kappa$ . Define the states and the query by:

$$\begin{aligned} \mathcal{S}_{\mathcal{A}(\iota \ddagger \kappa)} &\stackrel{\text{df.}}{=} \{n_{j_1 1}^{(l)} \dots n_{j_1 1}^{(1)} m_{i_k 0}^{(k)} \dots m_{i_1 0}^{(1)} | m_{i_k}^{(k)} \dots m_{i_1}^{(1)} \in \mathcal{S}_{\mathcal{A}(\iota)}, n_{j_1}^{(l)} \dots n_{j_1}^{(1)} \in \mathcal{S}_{\mathcal{A}(\kappa)}\} \\ \mathcal{Q}_{\mathcal{A}(\iota \ddagger \kappa)} : m_{i_0} &\mapsto \mathcal{Q}_{\mathcal{A}(\iota)}(m_i) \mid n_{j_1} \mapsto \mathcal{Q}_{\mathcal{A}(\kappa)}(n_j). \end{aligned}$$

Now, define the finite partial function  $\mathcal{A}(\iota \ddagger \kappa)_{n_{j_1 1}^{(l)} \dots n_{j_1 1}^{(1)} m_{i_k 0}^{(k)} \dots m_{i_1 0}^{(1)}}$  as  $\mathcal{A}(\kappa)_{n_{j_1}^{(l)} \dots n_{j_1}^{(1)}}$  if  $k = 0$ , and  $\mathcal{A}(\iota)_{m_{i_k}^{(k)} \dots m_{i_1}^{(1)}}$  otherwise, where we again insert additional bits 0, 1 on the righthand side of internal tags of symbols in the table. Note that P-views in  $\iota \ddagger \kappa$  are those in  $\iota$  followed by those in  $\kappa$ ; therefore it is straightforward to see that  $\text{st}(\mathcal{A}(\iota \ddagger \kappa)) = \iota \ddagger \kappa$  holds.

Finally, let  $\varphi^\dagger : [!A_0]_{\langle e \rangle \# f} \multimap [!B_1]_{\langle e' \rangle \# \tilde{f}}$  be the promotion of a strategy  $\varphi : [!A_0]_{\langle e \rangle \# f} \multimap [B_1]_{\tilde{f}}$  with an algorithm  $\mathcal{A}(\varphi)$  that realizes  $\varphi$ . We define  $\mathcal{S}_{\mathcal{A}(\varphi^\dagger)} \stackrel{\text{df.}}{=} \mathcal{S}_{\mathcal{A}(\varphi)}$  and  $\mathcal{Q}_{\mathcal{A}(\varphi^\dagger)} \stackrel{\text{df.}}{=} \mathcal{Q}_{\mathcal{A}(\varphi)}$ . Then roughly, the partial function  $\mathcal{A}(\varphi^\dagger)_m$  for each  $m \in \mathcal{S}_{\mathcal{A}(\varphi^\dagger)}$  is obtained from  $\mathcal{A}(\varphi)_m$  in such a way that if P-moves  $[a_0]_{\langle e \rangle \# f}, [b_1]_{\tilde{f}}$  occur in a play for  $\varphi$ , and the corresponding play for  $\varphi^\dagger$  begins with an initial move  $[b'_1]_{\langle e' \rangle \# \tilde{f}}$ , then  $\varphi^\dagger$  makes the corresponding moves  $[a_0]_{\langle \langle e' \rangle \# \langle e \rangle \rangle \# f}, [b_1]_{\langle e' \rangle \# \tilde{f}}$  in that play. This is certainly possible by modifying the manipulation of outer tags by  $\mathcal{A}(\varphi)_m$  appropriately with the help of m-views as follows:

1. The calculation of the next internal element by  $\mathcal{A}(\varphi^\dagger)_m$  is the same as  $\mathcal{A}(\varphi)_m$  since  $\mathcal{A}(\varphi)$  is assumed to be *standard*. Below, we focus on the calculation of outer tags.
2. Duplicate the input/output pairs in  $\mathcal{A}(\varphi)_m$  involved in the calculation of the next internal element but replace the opening move  $[q!A \multimap !B]_3$  and the last moves  $[m]_3$  by  $[q\mathcal{T}]_3$  and  $[q!A \multimap !B]_2$ , respectively. Also, we “postpone”, by m-views,  $(\mathcal{A}(\varphi)_m)$ ’s calculation of the next outer tag until the additional symbol  $\langle e' \rangle$  has been read off. Since  $\mathcal{A}(\varphi^\dagger)_m$  “sees” whether the opening move is  $[q!A \multimap !B]_3$  or  $[q\mathcal{T}]_3$ , i.e., its input includes an opening move, the newly added computations will never be confused with the old ones. In this manner,  $\mathcal{A}(\varphi^\dagger)_m$  learns  $!A$  or  $!B$  the last and the next moves in the P-view respectively belong to.
3. If the last and the next moves in the P-view both belong to  $!A$  (resp.  $!B$ ), then their outer tags are of the form  $\langle \langle e' \rangle \# \langle e \rangle \rangle \# f, \langle \langle e' \rangle \# \langle \tilde{e} \rangle \rangle \# \tilde{f}$  (resp.  $\langle e' \rangle \# f, \langle e' \rangle \# \tilde{f}$ ), respectively. They respectively correspond to moves with the same internal elements and the outer tags  $\langle e \rangle \# f, \langle \tilde{e} \rangle \# \tilde{f}$  (resp.  $f, \tilde{f}$ ) in  $\mathcal{A}(\varphi)_m$ . Since  $\mathcal{A}(\varphi)$  is assumed to be *standard*, with the help of m-views, we may clearly modify  $(\mathcal{A}(\varphi)_m)$ ’s computation from  $\langle e \rangle \# f$  to  $\langle \tilde{e} \rangle \# \tilde{f}$  (resp. from

<sup>4</sup>I.e., when the underlying game is not the terminal game  $I$ .

$f$  to  $\tilde{f}$  in such a way that  $(\mathcal{A}(\varphi^\dagger)_m)$ 's corresponding computation is standard and maps  $\langle\langle e' \rangle\# \langle e \rangle\rangle\#f$  (resp.  $\langle e' \rangle\#f$ ) to  $\langle\langle e' \rangle\# \langle \tilde{e} \rangle\rangle\#\tilde{f}$  (resp.  $\langle e' \rangle\#\tilde{f}$ ) whatever  $e'$  is (roughly it first “copy-cats”  $\langle\langle e' \rangle\#$  (resp.  $\langle e' \rangle\#$ ), and then the m-view at this point tells it to simulate the computation  $\langle e \rangle\#f \mapsto \langle \tilde{e} \rangle\#\tilde{f}$  (resp.  $f \mapsto \tilde{f}$ ), inserting another  $\rangle$  between  $\rangle$  and  $\#$ ).

4. If the last and the next moves belong to  $!B$  and  $!A$ , respectively, then their outer tags are of the form  $\langle e' \rangle\#$ ,  $\langle\langle e' \rangle\# \langle e \rangle\rangle\#$ . Note that they correspond to moves with the same internal elements and the outer tags  $\epsilon$ ,  $\langle e \rangle\#$ , respectively, in  $\mathcal{A}(\varphi)_m$ . When  $\mathcal{A}(\varphi^\dagger)_m$  continues, it first adds the symbol  $\langle$ , and then “copy-cats”  $\langle e' \rangle\#$ . At this point, the m-view tells  $\mathcal{A}(\varphi^\dagger)_m$  to simulate the calculation of  $\langle e \rangle\#$  of  $\mathcal{A}(\varphi)_m$ , inserting another  $\rangle$  between  $\rangle$  and  $\#$ .

We do not give a formal description of  $\mathcal{A}(\varphi^\dagger)_m$  as it would be much more involved and hard to read; however, the above description should suffice to indicate how we may construct it. ■

► Example 3.1.10. Consider the tensor  $\text{succ} \otimes \text{pred} : [!N_{00}]_{\langle e \rangle\#} \otimes [!N_{10}]_{\langle e' \rangle\#} \multimap [N_{01}] \otimes [N_{11}]$ . A typical play by  $\text{succ} \otimes \text{pred}$  is as follows:

$$\begin{array}{c}
\frac{[!N_{00}]_{\langle e \rangle\#} \otimes [!N_{10}]_{\langle e' \rangle\#} \xrightarrow{\text{succ} \otimes \text{pred}} [N_{01}] \otimes [N_{11}]}{[q_{11}]} \\
\begin{array}{c} [q_{10}]_{\langle \rangle\#} \\ [\bullet_{10}]_{\langle \rangle\#} \\ [q_{10}]_{\langle \rangle\#} \end{array} \qquad [q_{01}] \\
[q_{00}]_{\langle \rangle\#} \qquad [q_{01}] \\
[b_{00}]_{\langle \rangle\#} \qquad [\bullet_{01}] \\
[b_{10}]_{\langle \rangle\#} \qquad [b_{11}] \\
\qquad [q_{01}] \\
\qquad [b_{01}]
\end{array}$$

Applying the construction described in the proof of Theorem 3.1.9, we construct an algorithm

$\mathcal{A}(\text{succ} \otimes \text{pred})$  by  $\mathcal{S}_{\mathcal{A}(\text{succ} \otimes \text{pred})} \stackrel{\text{df.}}{=} \{q_{01}, q_{11}\}$ ,  $\mathcal{Q}_{\mathcal{A}(\text{succ} \otimes \text{pred})}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_{01} \vee m = q_{11} \\ \perp & \text{otherwise} \end{cases}$ ,

and  $\mathcal{A}(\text{succ} \otimes \text{pred})_{q_{01}}$  (resp.  $\mathcal{A}(\text{succ} \otimes \text{pred})_{q_{11}}$ ) is obtained from  $\mathcal{A}(\text{succ})_{q_1}$  (resp.  $\mathcal{A}(\text{pred})_{q_1}$ ) by replacing symbols  $m_i$  with  $m_{0i}$  (resp.  $m_{1i}$ ). It is easy to see that  $\mathcal{A}(\text{succ} \otimes \text{pred})$  achieves the computation in the above diagram, and moreover  $\text{st}(\mathcal{A}(\text{succ} \otimes \text{pred})) = \text{succ} \otimes \text{pred}$  holds.

► Example 3.1.11. Consider the pairing  $\langle \text{succ}, \text{pred} \rangle : [!N_0]_{\langle e \rangle\#} \multimap [N_{10}] \& [N_{11}]$ , where note that the 0, 1 digits in the codomain differ from those in the case of  $\otimes$ . Its typical plays are as follows:

$$\begin{array}{c}
\frac{[!N_0]_{\langle e \rangle\#} \xrightarrow{\langle \text{succ}, \text{pred} \rangle} [N_{10}] \& [N_{11}]}{[q_{10}]} \qquad \frac{[!N_0]_{\langle e \rangle\#} \xrightarrow{\langle \text{succ}, \text{pred} \rangle} [N_{10}] \& [N_{11}]}{[q_{11}]} \\
\begin{array}{c} [q_0]_{\langle \rangle\#} \\ [b_0]_{\langle \rangle\#} \end{array} \qquad [\bullet_{10}] \\
\qquad [q_{10}] \\
\qquad [b_{10}] \qquad [b_{11}]
\end{array}$$

Again, as described in the proof of Theorem 3.1.9, we construct an algorithm  $\mathcal{A}(\langle \text{succ}, \text{pred} \rangle)$  by

$\mathcal{S}_{\mathcal{A}(\langle \text{succ}, \text{pred} \rangle)} \stackrel{\text{df.}}{=} \{q_{10}, q_{11}\}$ ,  $\mathcal{Q}_{\mathcal{A}(\langle \text{succ}, \text{pred} \rangle)}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_{10} \vee m = q_{11} \\ \perp & \text{otherwise} \end{cases}$ , and  $\mathcal{A}(\langle \text{succ}, \text{pred} \rangle)_{q_{10}}$

► Example 3.1.12. Consider the promotion  $\text{succ}^\dagger : [!N_0]_{\langle e \rangle \#} \multimap [!N_1]_{\langle e' \rangle \#}$ . Its typical play is as depicted in the following diagram:

Let us apply the construction in the proof of Theorem 3.1.9. The set  $\mathcal{S}_{\mathcal{A}(succ^\dagger)}$  of states and the query  $\mathcal{Q}_{\mathcal{A}(succ^\dagger)}$  are the same as those of  $\mathcal{A}(succ)$ . For the computation of the outer tag of the next move (i.e., when the opening move is  $[q_T]_3$  in  $\mathcal{G}(M_{N \Rightarrow N})^3 \Rightarrow \mathcal{G}(M_{N \Rightarrow N})$ ), recall, e.g., how  $\mathcal{A}(succ)_{q_1}^*$  computes the second move from the opening move in  $N \Rightarrow N$ :

21

Let us see how  $\mathcal{A}(succ^\dagger)_{q_1}^*$  computes the outer tag of the next move when the last move is an opening  $[q_1]\langle 2 \rangle_\#$  and the next move is  $[q_0]\langle \langle 2 \rangle_\# \rangle_\#$  in  $!N \multimap !N$ :

$$\begin{array}{c}
\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \xRightarrow{\mathcal{A}(succ^\dagger)_{q_1}^*} \mathcal{G}(M_{N \Rightarrow N})_3 \\
\hline
\begin{array}{c}
[q_{N \Rightarrow N}]_0 \\
[\square]_0
\end{array}
\begin{array}{c}
[q_{N \Rightarrow N}]_2 \\
[q_1]_2
\end{array}
\begin{array}{c}
[q_{N \Rightarrow N}]_2 \\
[q_1]_2
\end{array}
\begin{array}{c}
[\langle \rangle]_3 \\
[q\tau]_3
\end{array}
\end{array}$$

$$\begin{array}{c}
[q_{N \Rightarrow N}]_2 \\
[q_1]_2
\end{array}
\begin{array}{c}
[q_{N \Rightarrow N}]_2 \\
[q_1]_2
\end{array}
\begin{array}{c}
[\langle \rangle]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q\tau]_2 \\
[\langle \rangle]_2
\end{array}
\begin{array}{c}
[\langle \rangle]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q\tau]_2 \\
[\square]_2
\end{array}
\begin{array}{c}
[\square]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q\tau]_2 \\
[\square]_2
\end{array}
\begin{array}{c}
[\square]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q\tau]_2 \\
[\rangle]_2
\end{array}
\begin{array}{c}
[\rangle]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q\tau]_2 \\
[\#]_2
\end{array}
\begin{array}{c}
[\#]_3 \\
[q\tau]_3
\end{array}$$

$$\begin{array}{c}
[q_{N \Rightarrow N}]_2 \\
[q_1]_2
\end{array}
\begin{array}{c}
[\langle \rangle]_3 \\
[q\tau]_3 \\
[\rangle]_3 \\
[q\tau]_3 \\
[\rangle]_3 \\
[q\tau]_3 \\
[\#]_3 \\
[q\tau]_3 \\
[\checkmark]_3
\end{array}$$

It should be clear how  $\mathcal{A}(succ^\dagger)_{q_1}^*$  calculates the outer tag of the next move in other cases. In this way,  $\text{st}(\mathcal{A}(succ^\dagger)) = succ^\dagger$  in fact holds.

► Example 3.1.13. Consider the concatenation  $\text{succ}^\dagger \ddagger \text{pred} : ([!N_{00}]_{\langle e \rangle \#} \multimap [!N_{10}]_{\langle e' \rangle \#}) \ddagger ([!N_{01}]_{\langle e' \rangle \#} \multimap [N_{11}])$ . Its typical play is as follows:

$$\begin{array}{c}
 \frac{[!N_{00}]_{\langle e \rangle \#} \xrightarrow{\text{succ}^\dagger} [!N_{10}]_{\langle e' \rangle \#} \quad [!N_{01}]_{\langle e' \rangle \#} \xrightarrow{\text{pred}} [N_{11}]}{[q_{11}]} \\
 \\
 \begin{array}{ccc}
 & [q_{10}]_{\langle \rangle \#} & [q_{01}]_{\langle \rangle \#} \\
 [q_{00}]_{\langle \rangle \# \langle \rangle \#} & & \\
 [b_{00}]_{\langle \rangle \# \langle \rangle \#} & & 
 \end{array} \\
 \\
 \begin{array}{ccc}
 & [\bullet_{10}]_{\langle \rangle \#} & [\bullet_{01}]_{\langle \rangle \#} \\
 & & [q_{01}]_{\langle \rangle \#} \\
 [q_{10}]_{\langle \rangle \#} & & \\
 [b_{10}]_{\langle \rangle \#} & & 
 \end{array} \\
 \\
 \begin{array}{ccc}
 & [b_{01}]_{\langle \rangle \#} & \\
 & & [b_{11}]
 \end{array}
 \end{array}$$

Applying the recipe in the proof of Theorem 3.1.9, we define an algorithm  $\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})$  as follows. Define  $\mathcal{S}_{\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})} \stackrel{\text{df.}}{=} \{q_{10}, q_{11}\}$ ,  $\mathcal{Q}_{\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_{10} \vee m = q_{11} \\ \perp & \text{otherwise} \end{cases}$ , and  $\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})_{q_{11}q_{10}}$  (resp.  $\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})_{q_{11}}$ ) is obtained from  $\mathcal{A}(\text{succ}^\dagger)_{q_1}$  (resp.  $\mathcal{A}(\text{pred})_{q_1}$ ) just by changing symbols  $m_i$  into  $m_{i0}$  (resp.  $m_{i1}$ ) in its finite table. It is then clear that  $\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})$  achieves the computation in the above diagram, and  $\text{st}(\mathcal{A}(\text{succ}^\dagger \ddagger \text{pred})) = \text{succ}^\dagger \ddagger \text{pred}$  holds.

## 3.2 Examples of atomic strategies

This section presents various examples of *consistent* and *effective* “atomic” strategies. Let us remark beforehand that these strategies except the *fixed-point strategies*  $\text{fix}_A$  are representable by finite view functions; thus, we need the notion of *effective strategies* only for promotion and  $\text{fix}_A$ .

► Remark. When describing strategies below, we usually keep justifiers implicit for brevity as they are always obvious in our examples.

► Example 3.2.1. Similarly to *zero* and *succ*, we may give an algorithm  $\mathcal{A}(\text{pred})$  for the *predecessor strategy*  $\text{pred} : [N_0]_{\langle e \rangle \#} \Rightarrow [N_1]$  defined by:

$$\begin{aligned}
 \text{pred} \stackrel{\text{df.}}{=} & \text{pref}(\{[q_1][q_0]_{\langle \rangle \#}[\bullet_0]_{\langle \rangle \#}[q_0]_{\langle \rangle \#}([\bullet_0]_{\langle \rangle \#}[\bullet_1][q_1][q_0]_{\langle \rangle \#})^n[b_0]_{\langle \rangle \#}[b_1] \mid n \in \mathbb{N}\} \\
 & \cup \{[q_1][q_0]_{\langle \rangle \#}[b_0]_{\langle \rangle \#}[b_1]\})^{\text{even}}
 \end{aligned}$$

whose states and query are the same as those for *succ*. At this point, it suffices to show a similar diagram for  $\mathcal{A}(\text{pred})_{q_1}^*$  as it is clear that there is a finite table  $\mathcal{A}(\text{pred})_{q_1}$  achieving it:

$$\begin{array}{c}
 \frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xrightarrow{\mathcal{A}(\text{pred})_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{N \Rightarrow N}]_3} \\
 \\
 \begin{array}{ccc}
 & [q_{N \Rightarrow N}]_2 & \\
 & [q_1]_2 ([b_0]_2) & \\
 & & [q_0]_3 ([b_1]_3)
 \end{array}
 \end{array}$$

$$\begin{array}{c}
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(\text{pred})_{q_1}^*} \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{N \Rightarrow N}]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [\bullet_0]_2 \end{array} \\
[q_1]_0 \in \text{Init (otherwise)} \qquad [q_0]_3 ([\bullet_1]_3) \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(\text{pred})_{q_1}^*} \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{\mathcal{T}}]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [b_0]_2 ([q_1]_2) \end{array} \\
\begin{array}{c} [\checkmark]_3 ([\langle \rangle]_3) \\ ([q_{\mathcal{T}}]_3) \\ ([\rangle]_3) \\ ([q_{\mathcal{T}}]_3) \\ ([\#]_3) \\ ([q_{\mathcal{T}}]_3) \\ ([\checkmark]_3) \end{array} \\
\frac{\mathcal{G}(M_{N \Rightarrow N})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow N})_2 \quad \xRightarrow{\mathcal{A}(\text{pred})_{q_1}^*} \mathcal{G}(M_{N \Rightarrow N})_3}{[q_{\mathcal{T}}]_3} \\
\begin{array}{c} [q_{N \Rightarrow N}]_2 \\ [\bullet_0]_2 \end{array} \\
([q_1]_0 \in \text{Init}) \text{ otherwise} \qquad \begin{array}{c} ([\langle \rangle]_3) [\checkmark]_3 \\ ([q_{\mathcal{T}}]_3) \\ ([\rangle]_3) \\ ([q_{\mathcal{T}}]_3) \\ ([\#]_3) \\ ([q_{\mathcal{T}}]_3) \\ ([\checkmark]_3) \end{array}
\end{array}$$

where  $[q_1]_0 \in \text{Init}$  (resp.  $[q_1]_0 \notin \text{Init}$ ) denotes the move  $[q_1]_0$  that is (resp. not) an initial occurrence. Note that it is “effectively computable” to decide whether an occurrence of a move is initial since it suffices to see if it has a pointer. Clearly  $\text{st}(\mathcal{A}(\text{pred})) = \text{pred}$ . Note that  $\mathcal{A}(\text{zero})$ ,  $\mathcal{A}(\text{succ})$  and  $\mathcal{A}(\text{pred})$  are all standard, and  $\text{zero}$ ,  $\text{succ}$  and  $\text{pred}$  are all trivially consistent.

► Example 3.2.2. For each game  $A$ , we may give an algorithm  $\mathcal{A}(cp_A)$  that realizes the copy-cat strategy  $cp_A : [A_0]_e \multimap [A_1]_e$  by  $\mathcal{Q}_{cp_A}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } \star \vdash_A [m] \\ \perp & \text{otherwise} \end{cases}$ ,  $\mathcal{S}_{cp_A} \stackrel{\text{df.}}{=} \{m \mid \star \vdash_A [m]\}$ ,

$|\mathcal{A}(cp_A)_m| \stackrel{\text{df.}}{=} 3$  for all  $m \in \mathcal{S}_{cp_A}$ , and

$$\begin{aligned}
\mathcal{A}(cp_A)_m : [q_{A \multimap A}]_3 \mapsto [q_{A \multimap A}]_2 \mid [q_{A \multimap A}]_3 [q_{A \multimap A}]_2 [a_0]_2 \mapsto [a_1]_3 \mid [q_{A \multimap A}]_3 [q_{A \multimap A}]_2 [a_1]_2 \mapsto [a_0]_3 \mid \\
[q_{\mathcal{T}}]_3 \mapsto [q_{\mathcal{T}}]_2 \mid [x]_3 [x]_2 [y]_2 \mapsto [y]_3 \mid [x]_2 [x]_3 [y]_3 \mapsto [y]_2
\end{aligned}$$

for all  $m \in \mathcal{S}_{cp_A}$ , where  $a \in \pi_1(M_A)$ ,  $x, y \in \pi_1(M_{\mathcal{G}(\mathcal{T})})$ . Accordingly,  $\mathcal{A}(cp_A)_m^*$  is as depicted in the following diagrams:



$$\begin{array}{c}
\frac{\mathcal{G}(M_{A \multimap A})_0 \quad \& \quad \mathcal{G}(M_{A \multimap A})_1 \quad \& \quad \mathcal{G}(M_{A \multimap A})_2 \quad \xRightarrow{\mathcal{A}(cp_A)^*_m} \quad \mathcal{G}(M_{A \multimap A})_3}{[q_{A \multimap A}]_3} \\
\begin{array}{c} [q_{A \multimap A}]_2 \\ [a_0]_2 \ ([a_1]_2) \end{array} \qquad \begin{array}{c} [a_1]_3 \ ([a_0]_3) \end{array} \\
\\
\frac{\mathcal{G}(M_{A \multimap A})_0 \quad \& \quad \mathcal{G}(M_{A \multimap A})_1 \quad \& \quad \mathcal{G}(M_{A \multimap A})_2 \quad \xRightarrow{\mathcal{A}(cp_A)^*_m} \quad \mathcal{G}(M_{A \multimap A})_3}{[q\tau]_3} \\
\begin{array}{c} [q\tau]_2 \\ [e]_2 \end{array} \qquad \begin{array}{c} [e]_3 \end{array} \\
\qquad \qquad \qquad \vdots \\
\begin{array}{c} [q\tau]_2 \\ [e']_2 \end{array} \qquad \begin{array}{c} [q\tau]_3 \\ [e']_3 \end{array} \\
\begin{array}{c} [q\tau]_2 \\ [\checkmark]_2 \end{array} \qquad \begin{array}{c} [q\tau]_3 \\ [\checkmark]_3 \end{array}
\end{array}$$

Then it is straightforward to see that  $\text{st}(\mathcal{A}(cp_A)) = cp_A$  holds, showing the effectivity of  $cp_A$ . Also,  $cp_A$  is trivially consistent, and  $\mathcal{A}(cp_A)$  is clearly standard. In a completely analogous way, we may show that the dereliction  $der_A : A \Rightarrow A$  for each game  $A$  is consistent and effective with a standard algorithm realizing it as well.

► **Example 3.2.3.** Consider the *case strategy*  $case_A : [A_0]_{\langle e'' \rangle \# f} \Rightarrow [A_{01}]_{\langle e' \rangle \# f} \Rightarrow [2_{011}]_{\langle e \rangle \#} \Rightarrow [A_{111}]_f$  on each game  $A$  defined by

$$\begin{aligned}
case_A \stackrel{\text{df.}}{=} & \text{pref}(\{[a_{111}][q_{011}]_{\langle \rangle \#} [\top_{011}]_{\langle \rangle \#} [a_0]_{\langle \rangle \#} . s \in P_{A \Rightarrow A \Rightarrow 2 \Rightarrow A} \mid [a_{111}][a_0]_{\langle \rangle \#} . s \in der_A^0\} \\
& \cup \{[a_{111}][q_{011}]_{\langle \rangle \#} [\perp_{011}]_{\langle \rangle \#} [a_{01}]_{\langle \rangle \#} . t \in P_{A \Rightarrow A \Rightarrow 2 \Rightarrow A} \mid [a_{111}][a_{01}]_{\langle \rangle \#} . t \in der_A^{01}\})^{\text{even}}
\end{aligned}$$

where  $der_A^0 : [A_0]_{\langle e'' \rangle \# f} \Rightarrow [A_{111}]_f$ ,  $der_A^{01} : [A_{01}]_{\langle e' \rangle \# f} \Rightarrow [A_{111}]_f$  are the same as the usual dereliction  $der_A : [A_0]_{\langle e' \rangle \# f} \Rightarrow [A_1]_f$  up to tags. Since this strategy distinguishes different copies of symbols  $|\#, \langle, \rangle$ , we explicitly write subscripts  $\alpha \in \{0, 1\}^*$  on them. We give an algorithm  $\mathcal{A}(case_A)$  that realizes  $case_A$  whose states and query are the same as  $\mathcal{A}(cp_A)$ , and for all  $m \in \mathcal{S}_{\mathcal{A}(case_A)}$  the instruction strategy  $\mathcal{A}(case_A)^*_m$  is as follows (again, since we have described  $cp_A$ , we skip formally writing down  $\mathcal{A}(case_A)_m$  as it should be clear):

$$\begin{array}{c}
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow 2 \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow 2 \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow 2 \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(case_A)^*_m} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow 2 \Rightarrow A})_3}{[q_{A \Rightarrow A \Rightarrow 2 \Rightarrow A}]_3} \\
\begin{array}{c} [q_{A \Rightarrow A \Rightarrow 2 \Rightarrow A}]_2 \\ [a_{111}]_2 \end{array} \qquad \begin{array}{c} [q_{011}]_3 \end{array} \\
\begin{array}{c} [q_{A \Rightarrow A \Rightarrow 2 \Rightarrow A}]_0 \\ [\square]_0 \end{array}
\end{array}$$

$$\begin{array}{c}
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3}{[qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_3} \\
\begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\ [a_{111}]_0 \end{array} \qquad \begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_2 \\ [\top_{011}]_2 \ (\ [\perp_{011}]_2 ) \end{array} \qquad [a_0]_3 \ ([a_{01}]_3) \\
\\
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3}{[qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_3} \\
\begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\ [a_0]_2 \ ([a_{01}]_2) \end{array} \qquad [a_{111}]_3 \\
\\
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3}{[qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_3} \\
\begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\ [a'_0]_0 \ ([a'_{01}]_0) \end{array} \qquad \begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_2 \\ [a_{111}]_2 \end{array} \qquad [a_0]_3 \ ([a_{01}]_3) \\
\\
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3}{[q\tau]_3} \\
\begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\ [\Box]_0 \end{array} \qquad \begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_2 \\ [a_{111}]_2 \end{array} \qquad \begin{array}{l} [\langle 011 \rangle]_3 \\ [q\tau]_3 \\ [\rangle_{011}]_3 \\ [q\tau]_3 \\ [\#_{011}]_3 \\ [q\tau]_3 \\ [\checkmark]_3 \end{array} \\
\\
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3}{[q\tau]_3} \\
\begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\ [\Box]_0 \end{array} \qquad \begin{array}{l} [qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_2 \\ [\top_{011}]_2 \ (\ [\perp_{011}]_2 ) \end{array} \qquad \begin{array}{l} [\langle 0 \rangle]_3 \ (\ [\langle 01 \rangle]_3 ) \\ [q\tau]_3 \\ [\rangle_0]_3 \ (\ [\rangle_{01} \rangle]_3 ) \\ [q\tau]_3 \\ [\#_0]_3 \ (\ [\#_{01} \rangle]_3 ) \\ [q\tau]_3 \\ [\checkmark]_3 \end{array}
\end{array}$$

$$\begin{array}{ccccccc}
\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 & \& \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 & \& \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 & \xRightarrow{\mathcal{A}(\text{case}_A)^*_m} & \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3 \\
\hline
& & & & [q_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A}]_2 & & [q\tau]_3 \\
& & & & [a_0]_2 ([a_{01}]_2) & & \\
& & & & [q\tau]_2 & & \\
& & & & [\langle 0 \rangle]_2 ([\langle 01 \rangle]_2) & & \\
& & & & [q\tau]_2 & & \\
& & & & [ \rangle_0 ]_2 ([ \rangle_{01} ]_2) & & \\
& & & & [q\tau]_2 & & \\
& & & & [\#_0]_2 ([\#_{01}]_2) & & \\
& & & & [q\tau]_2 & & \\
& & & & [e_0]_2 ([e_{01}]_2) & & [e_{111}]_3 \\
& & & & & & [q\tau]_3 \\
& & & & [q\tau]_2 & & \\
& & & & [e'_0]_2 ([e'_{01}]_2) & & [e'_{111}]_3 \\
& & & & & & \\
& & & & & \vdots & \\
& & & & & & [q\tau]_3 \\
& & & & [q\tau]_2 & & \\
& & & & [e''_0]_2 ([e''_{01}]_2) & & [e''_{111}]_3 \\
& & & & & & [q\tau]_3 \\
& & & & [q\tau]_2 & & \\
& & & & [\checkmark]_2 & & [\checkmark]_3
\end{array}$$

$$\begin{array}{c}
\mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{case}_A)^*_{\mathbf{m}}} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A})_3 \\
\hline
\begin{array}{c}
[qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_0 \\
[a'_0]_0 ([a'_{01}]_0)
\end{array}
\qquad
\begin{array}{c}
[qA \Rightarrow A \Rightarrow \mathbf{2} \Rightarrow A]_2 \\
[a_{111}]_2
\end{array}
\qquad
\begin{array}{c}
[q\tau]_3 \\
[\langle 0 \rangle_3 ([\langle 01 \rangle_3]) \\
[q\tau]_3 \\
[\rangle_0]_3 ([\rangle_{01}]_3) \\
[q\tau]_3 \\
[\#_0]_3 ([\#_{01}]_3) \\
[q\tau]_3 \\
[(e_1)_0]_3 ([e_{01}]_3) \\
[q\tau]_3 \\
[e'_0]_3 ([e'_{01}]_3) \\
\vdots \\
[q\tau]_3 \\
[e''_0]_3 ([e''_{01}]_3) \\
[q\tau]_3 \\
[\checkmark]_3
\end{array}
\end{array}$$

Clearly,  $\text{st}(\mathcal{A}(\text{case}_A)) = \text{case}_A$ , and so  $\text{case}_A$  is effective. And again,  $\mathcal{A}(\text{case}_A)$  is clearly standard, and  $\text{case}_A$  is trivially consistent.

► **Example 3.2.4.** Consider the *ifzero strategy*  $\text{zero?} : [N_0]_{\langle e_0 \rangle \#} \Rightarrow [\mathbf{2}_1]$  defined by  $\text{zero?} \stackrel{\text{df.}}{=} \text{pref}(\{[q_1][q_0]_{\langle \rangle \#}[b_0]_{\langle \rangle \#}[\top_1], [q_1][q_0]_{\langle \rangle \#}[\bullet_0]_{\langle \rangle \#}[\perp_1], \})^{\text{even}}$ , which is trivially consistent. Let us give an algorithm  $\mathcal{A}(\text{zero?})$  that realizes  $\text{zero?}$  as follows. Define  $\mathcal{Q}_{\mathcal{A}(\text{zero?})}(m) \stackrel{\text{df.}}{=} \begin{cases} \top & \text{if } m = q_1 \\ \perp & \text{otherwise} \end{cases}$ ,  $\mathcal{S}_{\mathcal{A}(\text{zero?})} \stackrel{\text{df.}}{=} \{q_1\}$ ,  $|\mathcal{A}(\text{zero?})_{q_1}| \stackrel{\text{df.}}{=} 3$ , and the instruction strategy  $\mathcal{A}(\text{zero?})^*_{q_1}$  is as depicted in the following diagrams (again, we omit the formal description of  $\mathcal{A}(\text{zero?})_{q_1}$  as it should be clear at this point):

$$\begin{array}{c}
\mathcal{G}(M_{N \Rightarrow \mathbf{2}})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_2 \quad \xRightarrow{\mathcal{A}(\text{zero?})^*_{q_1}} \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_3 \\
\hline
\begin{array}{c}
[qN \Rightarrow \mathbf{2}]_2 \\
[q_1]_2
\end{array}
\qquad
\begin{array}{c}
[qN \Rightarrow \mathbf{2}]_3 \\
[q_0]_3
\end{array}
\end{array}$$

$$\begin{array}{c}
\frac{\mathcal{G}(M_{N \Rightarrow \mathbf{2}})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_2 \quad \xRightarrow{\mathcal{A}(\text{zero?})_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_3}{[q\tau]_3} \\
\begin{array}{c} [q_{N \Rightarrow \mathbf{2}}]_2 \\ [q_1]_2 \end{array} \qquad \begin{array}{c} [\langle \rangle]_3 \\ [q\tau]_3 \\ [\rangle]_3 \\ [q\tau]_3 \\ [\sharp]_3 \\ [q\tau]_3 \\ [\checkmark]_3 \end{array} \\
\\
\frac{\mathcal{G}(M_{N \Rightarrow \mathbf{2}})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_2 \quad \xRightarrow{\mathcal{A}(\text{zero?})_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_3}{[q_{N \Rightarrow \mathbf{2}}]_3} \\
\begin{array}{c} [q_{N \Rightarrow \mathbf{2}}]_2 \\ [b_0]_2 \quad ([\bullet_0]_2) \end{array} \qquad [\top_1]_3 \quad ([\perp_1]_3) \\
\\
\frac{\mathcal{G}(M_{N \Rightarrow \mathbf{2}})_0 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_1 \quad \& \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_2 \quad \xRightarrow{\mathcal{A}(\text{zero?})_{q_1}^*} \quad \mathcal{G}(M_{N \Rightarrow \mathbf{2}})_3}{[q\tau]_3} \\
\begin{array}{c} [q_{N \Rightarrow \mathbf{2}}]_2 \\ [b_0]_2 \quad ([\bullet_0]_2) \end{array} \qquad [\checkmark]_3
\end{array}$$

We clearly have  $\text{st}(\mathcal{A}(\text{zero?})) = \text{zero?}$ , and  $\mathcal{A}(\text{zero?})$  is standard.

► **Example 3.2.5.** Consider the *fixed-point strategy*  $\text{fix}_A : ([A_{00}]_{\langle e' \rangle \sharp \langle e \rangle \sharp f} \Rightarrow [A_{10}]_{\langle e' \rangle \sharp f} \Rightarrow [A_1]_f$  for each game  $A$  [AJM00, HO00, McC98]. We have already described  $\text{fix}_A$  informally; here we give a more detailed account, but again, it should suffice to just give diagrams for  $\mathcal{A}(\text{fix}_A)_m^*$  ( $m \in \mathcal{S}_{\text{fix}_A}$ ):

$$\begin{array}{c}
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{fix}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3}{[q_{A \Rightarrow A \Rightarrow A}]_3} \\
\begin{array}{c} [q_{A \Rightarrow A \Rightarrow A}]_2 \\ [a_{00}]_2 \quad ([a_1]_2) \end{array} \qquad [a_{10}]_3 \\
\\
\frac{\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{fix}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3}{[q_{A \Rightarrow A \Rightarrow A}]_3} \\
\begin{array}{c} [q_{A \Rightarrow A \Rightarrow A}]_2 \\ [a_{10}]_2 \end{array} \qquad [a_1]_3 \quad ([a_{00}]_3) \\
\\
\begin{array}{c} [q_{A \Rightarrow A \Rightarrow A}]_0 \\ [a'_1]_0 \quad ([a'_{00}]_0) \end{array} \qquad [a_1]_3 \quad ([a_{00}]_3)
\end{array}$$

$$\begin{array}{c}
\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{fix}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3 \\
\hline
\begin{array}{ccc}
& & [q\tau]_3 \\
& [qA \Rightarrow A \Rightarrow A]_2 & \\
& [a_1]_2 & \\
& & [\langle \rangle]_3 \\
& & [q\tau]_3 \\
& & [ ]_3 \\
& & [q\tau]_3 \\
& & [\#]_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e]_2 & [e]_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e']_2 & [e']_3 \\
& & \vdots \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e'']_2 & [e'']_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\checkmark]_2 & [\checkmark]_3
\end{array}
\end{array}$$

$$\begin{array}{c}
\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(fix_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3 \\
\hline
\begin{array}{ccc}
& [q_{A \Rightarrow A \Rightarrow A}]_2 & [q\tau]_3 \\
& [a_{10}]_2 & \\
& [q\tau]_2 & \\
& [\langle \rangle]_2 & \\
& [q\tau]_2 & \\
& [ \rangle ]_2 & \\
& [q\tau]_2 & \\
& [\sharp]_2 & \\
& [q\tau]_2 & \\
& [e]_2 & \\
& & [e]_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e']_2 & [e']_3 \\
& & \vdots \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e'']_2 & [e'']_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\checkmark]_2 & [\checkmark]_3
\end{array}
\end{array}$$

$$\begin{array}{c}
\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1 \quad \& \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2 \quad \xRightarrow{\mathcal{A}(\text{fix}_A)_m^*} \quad \mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3 \\
\hline
\begin{array}{ccc}
& & [q\tau]_3 \\
& [qA \Rightarrow A \Rightarrow A]_2 & \\
& [a_{10}]_2 & \\
& [q\tau]_2 & \\
& [\langle \rangle_2 @ 0] & \\
& [q\tau]_2 & \\
& [\langle \rangle_2 @ 1] & \\
& & [\langle \rangle_3 @ 2] \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\tilde{e}]_2 & \\
& & [\tilde{e}]_3 \\
& & \vdots \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\rangle]_2 @ 1 & \\
& & [\rangle]_3 @ 2 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\#]_2 & \\
& & [\#]_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\langle \rangle_2 @ 3] & \\
& & [\langle \rangle_3 @ 4] \\
& & \vdots \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\rangle]_2 @ 3 & \\
& & [\rangle]_3 @ 4 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\rangle]_2 @ 0 & \\
& [q\tau]_2 & \\
& [e]_2 & \\
& & [e]_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e']_2 & \\
& & [e']_3 \\
& & \vdots \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [e'']_2 & \\
& & [e'']_3 \\
& & [q\tau]_3 \\
& [q\tau]_2 & \\
& [\checkmark]_2 & \\
& & [\checkmark]_3
\end{array}
\end{array}$$



$\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_0$	$\&$	$\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_1$	$\&$	$\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_2$	$\xRightarrow{\mathcal{A}(fix_A)_m^*}$	$\mathcal{G}(M_{A \Rightarrow A \Rightarrow A})_3$
				$[q_{A \Rightarrow A \Rightarrow A}]_2$		$[q\tau]_3$
				$[a_{00}]_2$		$[\langle \rangle]_3 @0$
				$[q\tau]_2$		$[q\tau]_3$
				$[\langle \rangle]_2 @1$		$[\langle \rangle]_3 @2$
				$[q\tau]_2$		$[q\tau]_3$
				$[\tilde{e}]_2$		$[\tilde{e}]_3$
					$\vdots$	
				$[q\tau]_2$		$[q\tau]_3$
				$[\rangle]_2 @1$		$[\rangle]_3 @2$
				$[q\tau]_2$		$[q\tau]_3$
				$[\#]_2$		$[\#]_3$
				$[q\tau]_2$		$[q\tau]_3$
				$[\langle \rangle]_2 @3$		$[\langle \rangle]_3 @4$
					$\vdots$	
				$[q\tau]_2$		$[q\tau]_3$
				$[\rangle]_2 @3$		$[\rangle]_3 @4$
						$[q\tau]_3$
				$[q\tau]_2$		$[\rangle]_3 @0$
				$[e]_2$		$[q\tau]_3$
						$[e]_3$
				$[q\tau]_2$		$[q\tau]_3$
				$[e']_2$		$[e']_3$
					$\vdots$	
				$[q\tau]_2$		$[q\tau]_3$
				$[e'']_2$		$[e'']_3$
						$[q\tau]_3$
				$[q\tau]_2$		
				$[\surd]_2$		$[\surd]_3$

where  $@i$  in the diagrams indicates the pairs  $\langle, \rangle$  of mates, i.e.,  $\langle$  and  $\rangle$  with the same  $i$  form such a pair. Note that with m-views, there is an obvious finite table  $\mathcal{A}(\text{fix}_A)_m$  that implements the instruction strategy  $\mathcal{A}(\text{fix}_A)_m^*$ . It is then not hard to see that  $\text{st}(\mathcal{A}(\text{fix}_A)) = \text{fix}_A$  holds, showing that  $\text{fix}_A$  is effective. Also, it is easy to see that  $\text{fix}_A$  is consistent, and  $\mathcal{A}(\text{fix}_A)$  is standard.

### 3.3 Turing completeness

In the previous sections, we have seen that every “atomic” strategy that is *definable* in the language PCF [AM99] is consistent and realized by a standard algorithm, and constructions on strategies preserve this property. From this, our main theorem immediately follows:

► **Theorem 3.3.1** (Main theorem). *Every (static) strategy  $\sigma : A$  definable in PCF [AM99] has a consistent and effective strategy  $\phi_\sigma : D_A$  that satisfies  $\mathcal{H}^\omega(\phi_\sigma) = \sigma \wedge \mathcal{H}^\omega(D_A) \trianglelefteq A$  up to tags.*

*Proof.* First, see [AM99] for the strategies definable in the language PCF, (and [YA16] for the (implicitly) underlying bicategory  $\mathcal{DG}$  of dynamic games and strategies though it is not strictly necessary). We have already shown in the previous sections that the “atomic” strategies such as  $\text{der}_A$ ,  $\text{succ}$ ,  $\text{pred}$ ,  $\text{zero}?$ ,  $\text{case}_A$ ,  $\text{fix}_A$  are all consistent and effective, in particular realized by standard algorithms. Note that *projections* are derelictions up to internal tags, and so they are clearly consistent and effective. Similarly, the *currying*  $\Lambda$  and *uncurrying*  $\Lambda^{-1}$  operations are just to modify internal tags; thus, they clearly preserve consistency and effectivity (in particular realizability by standard algorithms) of strategies. In particular, the *evaluation strategies*  $\text{ev}_{A,B} : (A \Rightarrow B) \& A \rightarrow B$  are obtained from derelictions by uncurrying, and so they are consistent and effective as well.

Now, note that we may enumerate all the strategies definable in PCF by the following inductive construction of a set  $\mathcal{S}$  of strategies:

1.  $\sigma \in \mathcal{S}$  if  $\sigma : A$  is “atomic”
2.  $\Lambda(\sigma) \in \mathcal{S}$  if  $\sigma \in \mathcal{S}$  such that  $\sigma : G$  with  $\mathcal{H}^\omega(G) \trianglelefteq A \& B \rightarrow C$  for some games  $G, A, B, C$
3.  $\langle \varphi, \psi \rangle \in \mathcal{S}$  if  $\varphi, \psi \in \mathcal{S}$  such that  $\varphi : L, \psi : R$  with  $\mathcal{H}^\omega(L) \trianglelefteq C \rightarrow A, \mathcal{H}^\omega(R) \trianglelefteq C \rightarrow B$  for some games  $L, R, A, B, C$
4.  $\iota^\dagger; \kappa \in \mathcal{S}$  if  $\iota, \kappa \in \mathcal{S}$  such that  $\iota : J, \kappa : K$  with  $\mathcal{H}^\omega(J) \trianglelefteq A \rightarrow B, \mathcal{H}^\omega(K) \trianglelefteq B \rightarrow C$  for some games  $J, K, A, B, C$ .

We may assign a consistent and effective strategy  $\phi_\sigma$  to each  $\sigma \in \mathcal{S}$  as follows: 1.  $\phi_\sigma \stackrel{\text{df.}}{=} \sigma$  if  $\sigma$  is “atomic”; 2.  $\phi_{\Lambda(\sigma)} \stackrel{\text{df.}}{=} \Lambda(\phi_\sigma)$ ; 3.  $\phi_{\langle \varphi, \psi \rangle} \stackrel{\text{df.}}{=} \langle \phi_\varphi, \phi_\psi \rangle$ ; 4.  $\phi_{\iota^\dagger; \kappa} \stackrel{\text{df.}}{=} \phi_\iota^\dagger \ddagger \phi_\kappa$ .

By the above argument and Theorem 3.1.9,  $\phi_\sigma$  is in fact consistent and effective for all  $\sigma \in \mathcal{S}$ . It remains to show  $\mathcal{H}^\omega(\phi_\sigma) \cong \sigma$  for all  $\sigma \in \mathcal{S}$  (it is similar to show the subgame relation), where  $\cong$  denotes the equality up to tags. We may show it by induction with basic results in [YA16]:

1.  $\mathcal{H}^\omega(\phi_\sigma) = \mathcal{H}^\omega(\sigma) = \sigma$  if  $\sigma$  is “atomic” since every “atomic” strategy is static
2.  $\mathcal{H}^\omega(\phi_{\Lambda(\sigma)}) = \mathcal{H}^\omega(\Lambda(\phi_\sigma)) = \Lambda(\mathcal{H}^\omega(\phi_\sigma)) \cong \Lambda(\sigma)$  by the induction hypothesis
3.  $\mathcal{H}^\omega(\phi_{\langle \varphi, \psi \rangle}) = \mathcal{H}^\omega(\langle \phi_\varphi, \phi_\psi \rangle) = \langle \mathcal{H}^\omega(\phi_\varphi), \mathcal{H}^\omega(\phi_\psi) \rangle \cong \langle \varphi, \psi \rangle$  by the induction hypothesis
4.  $\mathcal{H}^\omega(\phi_{\iota^\dagger; \kappa}) = \mathcal{H}^\omega(\phi_\iota^\dagger \ddagger \phi_\kappa) = \mathcal{H}^\omega(\phi_\iota)^\dagger; \mathcal{H}^\omega(\phi_\kappa) \cong \iota^\dagger; \kappa$  by the induction hypothesis

which completes the proof. ■

Since PCF is *Turing complete* [Gun92, LN15], this result particularly implies the following:

► **Corollary 3.3.2** (Turing completeness). *Every partial recursive function  $f : \mathbb{N}^k \rightarrow \mathbb{N}$ , where  $k \in \mathbb{N}$ , has a consistent and effective strategy  $\phi_f : D_f$  such that  $\mathcal{H}^\omega(D_f) \trianglelefteq N^k \Rightarrow N$  and  $\underline{f(n)} \simeq \mathcal{H}^\omega(\langle \underline{n_1}, \underline{n_2}, \dots, \underline{n_k} \rangle^\dagger \ddagger \phi_f)$  up to tags for all  $(n_1, n_2, \dots, n_k) \in \mathbb{N}^k$ .*

## 4 Conclusion and future work

We have presented the first *intrinsic* notion of “effective computability” in game semantics. Due to its *semantic* and *non-inductive* nature, it can be seen as a fundamental investigation of the mathematical notion of “effective computation” beyond the classical computation.

There are many directions for further work; here we only mention some of them. First, we need to analyze the exact computational power of effective strategies, in comparison with other known notion of higher-order computability [LN15]. Also, as an application, the present framework may give an accurate measure for computational complexity [Koz06]. However, the most imminent future work is perhaps, by exploiting the flexibility of game semantics, to enlarge the scope of the present work (i.e., not only the language PCF) in order to establish a mathematical model of various (constructive) logics and programming languages.

**Acknowledgements.** The author acknowledges support from Funai Overseas Scholarship, and also he is grateful to Samson Abramsky and Robin Piedeleu for fruitful discussions.

## References

- [A<sup>+</sup>97] Samson Abramsky et al. Semantics of Interaction: An Introduction to Game Semantics. *Semantics and Logics of Computation, Publications of the Newton Institute*, pages 1–31, 1997.
- [Abr14] Samson Abramsky. Intensionality, Definability and Computation. In *Johan van Benthem on Logic and Information Dynamics*, pages 121–142. Springer, 2014.
- [AJ94] Samson Abramsky and Radha Jagadeesan. Games and Full Completeness for Multiplicative Linear Logic. *The Journal of Symbolic Logic*, 59(02):543–574, 1994.
- [AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full Abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000.
- [AM99] Samson Abramsky and Guy McCusker. Game Semantics. In *Computational logic*, pages 1–55. Springer, 1999.
- [B<sup>+</sup>84] Hendrik Pieter Barendregt et al. *The Lambda Calculus*, volume 3. North-Holland Amsterdam, 1984.
- [Chu36] Alonzo Church. An Unsolvability Problem of Elementary Number Theory. *American journal of mathematics*, 58(2):345–363, 1936.
- [Chu40] Alonzo Church. A Formulation of the Simple Theory of Types. *The journal of symbolic logic*, 5(02):56–68, 1940.
- [Cur30] Haskell B Curry. Grundlagen der kombinatorischen Logik. *American journal of mathematics*, 52(4):789–834, 1930.

- [Cut80] Nigel Cutland. *Computability: An Introduction to Recursive Function Theory*. Cambridge university press, 1980.
- [Gun92] Carl A Gunter. *Semantics of Programming Languages: Structures and Techniques*. MIT press, 1992.
- [Han94] Chris Hankin. *Lambda Calculi: A Guide for the Perplexed*. 1994.
- [HO00] J Martin E Hyland and C-HL Ong. On Full Abstraction for PCF: I, II, and III. *Information and computation*, 163(2):285–408, 2000.
- [Hyl97] Martin Hyland. Game Semantics. *Semantics and logics of computation*, 14:131, 1997.
- [Koz06] Dexter C Kozen. *Theory of Computation*. Springer Science & Business Media, 2006.
- [Koz12] Dexter C Kozen. *Automata and Computability*. Springer Science & Business Media, 2012.
- [LN15] John Longley and Dag Normann. *Higher-Order Computability*. Springer, 2015.
- [McC98] Guy McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Springer Science & Business Media, 1998.
- [Mit96] John C Mitchell. *Foundations for Programming Languages*, volume 1. MIT press Cambridge, 1996.
- [Nic94] Hanno Nickau. Hereditarily Sequential Functionals. In *Logical Foundations of Computer Science*, pages 253–264. Springer, 1994.
- [Plo77] Gordon D. Plotkin. Lcf considered as a Programming Language. *Theoretical computer science*, 5(3):223–255, 1977.
- [RR67] Hartley Rogers and H Rogers. *Theory of Recursive Functions and Effective Computability*, volume 5. McGraw-Hill New York, 1967.
- [Sch24] Moses Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92(3):305–316, 1924.
- [Tur36] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.
- [TVD14] Anne Sjerp Troelstra and Dirk Van Dalen. *Constructivism in Mathematics*, volume 2. Elsevier, 2014.
- [Win93] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT press, 1993.
- [YA16] Norihiro Yamada and Samson Abramsky. Dynamic Games and Strategies. *arXiv preprint arXiv:1601.04147*, 2016.